

FlexNet Code Insight 2018 R3

Installation & Configuration Guide

Legal Information

Book Name: FlexNet Code Insight 2018 R3 Installation and Configuration Guide
Part Number: FNCI-2018R3-IG00
Product Release Date: October 2018

Copyright Notice

Copyright © 2018 Flexera.

This publication contains proprietary and confidential information and creative works owned by Flexera and its licensors, if any. Any use, copying, publication, distribution, display, modification, or transmission of such publication in whole or in part in any form or by any means without the prior express written permission of Flexera is strictly prohibited. Except where expressly provided by Flexera in writing, possession of this publication shall not be construed to confer any license or rights under any Flexera intellectual property rights, whether by estoppel, implication, or otherwise.

All copies of the technology and related information, if allowed by Flexera, must display this notice of copyright and ownership in full.

Intellectual Property

For a list of trademarks and patents that are owned by Flexera, see <https://www.flexera.com/producer/company/about/intellectual-property/>. All other brand and product names mentioned in Flexera products, product documentation, and marketing materials are the trademarks and registered trademarks of their respective owners.

Restricted Rights Legend

The Software is commercial computer software. If the user or licensee of the Software is an agency, department, or other entity of the United States Government, the use, duplication, reproduction, release, modification, disclosure, or transfer of the Software, or any related documentation of any kind, including technical data and manuals, is restricted by a license agreement or by the terms of this Agreement in accordance with Federal Acquisition Regulation 12.212 for civilian purposes and Defense Federal Acquisition Regulation Supplement 227.7202 for military purposes. The Software was developed fully at private expense. All other use is prohibited.

Contents

1	Installing FlexNet Code Insight	9
	System Requirements	9
	Platform Support	10
	Database Support	10
	MySQL Required Components	10
	SQL Server Required Components	11
	Browser Support	11
	Recommended Hardware	12
	Deployment Models	12
	Configuration Guidelines	12
	Recommended Software	14
	Database Client	14
	Preparing to Install FlexNet Code Insight	14
	Setting Up the Database	14
	Setting Up the MySQL Database	14
	Sample Procedure for Creating an Appropriate Database Schema and User	15
	MySQL Database Settings	15
	Setting Up the SQL Server Database	18
	Phase 1: Install the SQL Server Instance	18
	Phase 2: Set Up the SQL Server Database	19
	Network and Firewall Considerations	20
	Setting the Open File Limit for Linux/Unix	20
	Enabling Secure HTTP Over SSL	21
	Enabling an HTTPS Connection	21
	Purchasing a Secure Site SSL certificate	22
	Creating a Keystore for a Purchased Secure Site SSL Certificate--Example	23
	Generating a Self-signed Certificate	23
	Using a Self-signed Certificate--Example	24
	Configuring a Networking Proxy Server Connection	24

Installing FlexNet Code Insight	25
Gathering the Required Files	25
Launching the Installer	25
Running FlexNet Code Insight as a Service	26
In a Windows Environment	26
In a Linux Environment	27
Starting & Stopping Tomcat	28
Opening FlexNet Code Insight	28
Uninstalling FlexNet Code Insight	29
Uninstalling on Windows	29
Uninstalling on Linux	30
Dropping the SQL Server Database	30
Contacting Support	30
 2 Configuring FlexNet Code Insight	 31
Creating or Editing a Scan Server	31
Managing Users	32
Creating or Editing Users	32
Finding Users	33
Disabling User Accounts	34
Setting the Electronic Update Frequency	34
Configuring an Email Server	35
Configuring LDAP	36
Synchronizing User Name Data	36
Setting Up a User Search Filter	36
Sample Search Query	37
Server Paging	37
User Authentication	37
LDAP over SSL	37
Implementing LDAP	37
Configuring FlexNet Code Insight to Use Single Sign-On	38
Prerequisite Tasks for Configuring Code Insight for SSO	38
Configure HTTPS on the FlexNet Code Insight Server	38
Set Up SSO Users	39
Configuring Code Insight for SSO	39
Step 1: Copy the Directory That Will Contain Provider Metadata	39
Step 2: Prepare the Environment Properties File	39
Step 3: Configure the SSO Common Properties File	40
Step 4: Customize the Sample Service Provider Metadata File	41
Step 5: Obtain the Identity Provider Metadata File	42
Log In Using SSO Credentials	42
Example Okta Setup for Code Insight SSO	42

Managing Scan Profiles	43
Creating or Editing Scan Profiles	43
Scan Profile Fields	44
About Scanning without the Compliance Library	45
Creating Exclusion Patterns for Scan Profiles	45
Setting Project Defaults	47
About FlexNet Code Insight Server REST APIs	50
3 Installing & Configuring FlexNet Code Insight Plugins	51
About Plugins	51
Generating a JWT Authorization Token	52
Downloading Plugins	53
The Jenkins Plugin	53
Prerequisite for the Jenkins Plugin	54
Setting Heap Size for the Jenkins Plugin	54
Setting Up the Code Insight Jenkins Plugin	55
Support for the Jenkins Pipeline	56
Providing the Pipeline Script for the Scan Step	56
Pipeline Code Examples for Running the Scan	56
<i>Example Declarative Pipeline Code to Run the Scan</i>	57
<i>Example Scripted Pipeline Code to Run the Scan</i>	57
The Scan Scheduler Plugin for Jenkins	58
The JFrog Artifactory Plugin	59
Prerequisites for the Artifactory Plugin	59
Installing the Artifactory Plugin	59
Scanning an Artifactory Repository Using a Cron Job	60
Scanning an Artifactory Repository Using REST API	60
Requirements When Using REST API to Scan Repositories	61
<i>Prerequisite for Scanning Repositories</i>	61
<i>Required Option When Using the “https” Protocol</i>	61
Scanning All Repositories	61
Scanning a Specific Repository	61
Reloading the Artifactory Plugin	62
Scan Results	62
The Docker Images Scan Plugin	62
Installing and Launching the Docker Images Plugin	62
The Bamboo Plugin	64
Installing & Configuring the Bamboo Plugin	64
The Maven Plugin	65
More About the Maven Scan Plugin	65
Prerequisites for the Maven Scan Plugin	65
Installing and Configuring the Maven Scan Plugin	66
Cleaning the Application Project	68
Running the Maven Goal for the Code Insight Scan	68

The Gradle Plugin	68
Installing and Configuring the Gradle Plugin	69
The Apache Ant Plugin	70
Configuring the Plugin	70
Executing the Scan	71
The Visual Studio Team Services (VSTS) Extension	72
Prerequisite	72
Installing the FlexNet Code Insight VSTS Extension	72
Adding a FlexNet Code Insight Scan Task to Your Agent Job	73
Scan Integration With Build Environments Through the Generic Scan Plugin	74
Downloading the Generic Scan Plugin	75
Prerequisites for Using the Generic Scan Plugin	75
The TeamCity Plugin	75
Prerequisites	76
Installing the Generic Scan Agent on TeamCity Agent Configured on Windows	76
Configuring a Build to Run a Code Insight Scan	76
Executing the Build	78
The GitLab Plugin	78
Prerequisites	78
Installing the Generic Scan Agent on GitLab Runner Configured on Windows	78
Configuring the CI/CD Pipeline to Run a Code Insight Scan	79
Executing the Build	80
4 Integrating with Source Code Management	81
Why Use Source Code Management (SCM)?	81
Configuring SCM	81
Prerequisites	82
SCM Command Line Client	82
Recommended Clients	82
Setting the Environment Variable	83
Git Protocol Options	83
Anonymous HTTP	84
Authenticated HTTP	84
SSH	84
SSH Over HTTP	86
Perforce Protocol Options	86
TFS Protocol and Credentials Configuration	86
HTTPS Protocol Support	86
Special Requirement for VSTS Projects in TFS	87
5 Integrating with Application Lifecycle Management	89
About Integration with Application Lifecycle Management (ALM) Systems	89
The Jira Plugin	89
Prerequisites for the Jira Plugin	90

Configuring the Jira Plugin 90

 Adding a Jira Instance 90

 Using Code Insight Variables 91

 Synchronizing Work Items 92

 Deleting an ALM Instance 93

Installing FlexNet Code Insight

This section contains the following topics covering the installation and startup of FlexNet Code Insight:

- [System Requirements](#)
- [Preparing to Install FlexNet Code Insight](#)
- [Installing FlexNet Code Insight](#)
- [Running FlexNet Code Insight as a Service](#)
- [Starting & Stopping Tomcat](#)
- [Opening FlexNet Code Insight](#)
- [Uninstalling FlexNet Code Insight](#)
- [Contacting Support](#)

System Requirements

Before installing FlexNet Code Insight, ensure that the following are installed on your system:

- A supported database instance and its associated connector. See [Database Support](#) for a description of supported databases and connectors.
- A FlexNet Code Insight license key file (codeinsight.key)
- On Linux machines, ensure that the number of open file handles is more than 50k, which is typically set with the `ulimit` command. For more information about the open file limit, see [Setting the Open File Limit for Linux/Unix](#).
- Any requirements specific to your FlexNet Code Insight plugin and remote data source.



Note • The JRE is included in the installation; a separate download is not necessary. Only JRE 8 is supported.

The following provides additional requirements:

- [Platform Support](#)
- [Database Support](#)
- [Browser Support](#)
- [Recommended Hardware](#)
- [Recommended Software](#)

Platform Support

FlexNet Code Insight supports the following platforms:

- Windows Server 2012
- Windows Server 2016
- RHEL 6.x, 7.x
- CentOS 6.x, 7.x

Database Support

FlexNet Code Insight requires that either a MySQL or SQL Server database be installed. The following lists components required to install and configure a database for use by Code Insight:

- [MySQL Required Components](#)
- [SQL Server Required Components](#)

MySQL Required Components

The following describes the components needed to install and run MySQL as the FlexNet Code Insight database:

- MySQL 5.7 community edition, downloaded from <https://dev.mysql.com/downloads/mysql/5.7.html>.



Note • *Code Insight does not support the Docker version of My SQL. (It supports the native version only.)*

- The JDBC driver connector file, `mysql-connector-java-5.1.x-bin.jar`. You can download this file from <http://dev.mysql.com/downloads/connector/j/5.1.html>.

This connector is required to enable FlexNet Code Insight to connect to the MySQL database.

- An environment that can support the required size settings listed in [MySQL Database Settings](#).

SQL Server Required Components

The following lists the required components needed to install and run SQL Server as the Code Insight database:

- SQL Server 2016 Sp2 (recommended version for best performance).
- The JDBC driver connector file, `mssql-jdbc-6.4.0.jre8.jar`. You can download this file from <https://www.microsoft.com/en-us/download/details.aspx?id=56615>.

This connector is required to enable Code Insight to connect to the SQL Server database.

- The package `sql_server_pre_install_scripts.zip` containing the scripts needed to set up the SQL Server database for Code Insight. See [Downloading the Scripts Needed to Set Up the SQL Server Database](#) for instructions on the download process.
- At least one disk (OS or non-OS) with 100 GB free space.

Downloading the Scripts Needed to Set Up the SQL Server Database

Use the following steps to download the package containing the script files needed to set up the SQL Server database for Code Insight.



Task

To download the package containing the scripts

1. Log into the Customer Community page of the Flexera website:
<https://flexeracommunity.force.com/customer/>
2. Click **Downloads**.
3. Click the **Access** button under **FlexNet Code Insight**. The Product and License Center page appears.
4. Select **FlexNet Code Insight** from the **Your Downloads** list.
5. Select the version of FlexNet Code Insight from the list. The **Downloads** page appears.
6. Download the `sql_server_pre_install_scripts.zip` file.
7. When the download finishes, extract the following files to a location accessible for later execution using the SQL Server console, as described in [Setting Up the SQL Server Database](#):
 - `palamida_serversettings.sql`
 - `palamida_db_creation_with_maintainenceplan.sql`

A third script, `palamida_db_drop_with_maintainenceplan.sql`, is used to drop the database and is *not* used as part of the database setup. Instructions for dropping the database are found in [Dropping the SQL Server Database](#).

Browser Support

FlexNet Code Insight supports the following browsers:

- Chrome (latest stable version)
- Internet Explorer (latest stable version)
- Firefox (latest stable version)



Note • FlexNet Code Insight no longer allows uppercase or mixed case when entering the application's URL. To start FlexNet Code Insight in a browser, you must enter **codeinsight** in lowercase.

Recommended Hardware

The recommended deployments and configurations are explained in this section:

- [Deployment Models](#)
- [Configuration Guidelines](#)

Deployment Models

The FlexNet Code Insight deployment model may be configured as a single-node or a multi-node deployment. Each deployment consists of the following elements:

Table 1-1 • Deployment Models

Entity	Description
Core Server	Main interface to FlexNet Code Insight.
Scan Server	Contains codebase to be scanned (required for local scans only, not required for remote scans) and the Compliance Library (CL), which is required for Exact and source code fingerprint (SCF) matching.
Database	Central database containing all library metadata supplied by electronic update and all stored scan results.

Configuration Guidelines

The following configurations are supported.



Note • For optimum performance, it is **highly** recommended that you use the Single Server Configuration, in which the Core Server, Scan Server, and Database are located on the same machine.

Table 1-2 • Supported Configurations

Configuration	CPU (Cores)	Memory	Disk Space
Single Server: Core Server Scan Server Database	2-CPU (each at least 2 GHZ+) with 8+ cores on the server	64 GB	Server: 500 GB High-speed Disk for the Database (SSD Recommended) 500 GB High-speed Disk for the Core/Scanner to store the codebase 1 TB SSD Disk for the Compliance Library (CL)
Server 1: Core/Scanner Server 2: Database	2-CPU (each at least 2 GHZ+) 8+ cores on each server	Server 1: 32 GB Server 2: 32 GB	Server 1: 500 GB High-speed Disk for Core/Scanner to store the codebase 1 TB SSD Disk for the Compliance Library (CL) Server 2: 500 GB High-speed Disk for the Database (SSD Recommended)
Server 1: Core Server 2: Scanner Server 3: Database	2-CPU (each at least 2 GHZ+) 8+ cores on each server	Server 1: 32 GB Server 2: 32 GB Server 3: 32GB	Server1: 250GB High-speed Disk for Core Server2: 500GB High-speed Disk for Core/Scanner to store the codebase 1 TB SSD Disk for the Compliance Library (CL) Server 3: 500 GB High-speed Disk for Core/Scanner to store the codebase



Note • A multi-scan deployment model is not available in this release.

Recommended Software

The following software is recommended for FlexNet Code Insight.

Database Client

A SQL client or command-line interface is necessary to run database scripts. The following free SQL clients are available:

- HeidiSQL: <http://www.heidisql.com/download.php>
- MySQL Workbench: <http://www.mysql.com/products/workbench/>

Preparing to Install FlexNet Code Insight

Installing FlexNet Code Insight is a simple, prompt-driven process, but before beginning the installation, you will need to do the following:

- Ensure that you have met the prerequisites in [System Requirements](#).
- Follow the procedure in [Setting Up the Database](#).
- Perform any additional environmental and communication configuration for Code Insight, such as the following:
 - [Network and Firewall Considerations](#)
 - [Setting the Open File Limit for Linux/Unix](#)
 - [Enabling Secure HTTP Over SSL](#)
 - [Configuring a Networking Proxy Server Connection](#)

Setting Up the Database

Before you install FlexNet Code Insight, a database administrator must set up the MySQL or SQL Server database for use by Code Insight:

- [Setting Up the MySQL Database](#)
- [Setting Up the SQL Server Database](#)

Setting Up the MySQL Database

The database administrator needs to perform the following steps to set up the MySQL database for FlexNet Code Insight.



Task **To set up the MySQL database for Code Insight:**

1. Install the MySQL instance.



Note • Installing the instance on a server other than the one on which Code Insight is installed might cause performance degradation.

2. Configure the database instance as described in [MySQL Database Settings](#).
3. Create a database schema (with a recommended name of *codeinsight*) and a user who has appropriate access privileges to access the database. The procedure described in [Sample Procedure for Creating an Appropriate Database Schema and User](#) can be used to perform these tasks.

Sample Procedure for Creating an Appropriate Database Schema and User

The following is a sample procedure that the database administrator can to create a Code Insight database schema and a database user.



Task **To create a database schema and user:**

1. At the command line, log into MySQL as the root user:

`mysql -u root -p`
2. Type the MySQL root password, and press **Enter**.
3. To create a database and user, type the following command, replacing the username (*fnciuser*) with the user you want to create, and replace *Fnci%1234* with the user's password:

```
CREATE DATABASE codeinsight;  
CREATE USER fnciuser IDENTIFIED BY 'Fnci%1234';  
GRANT ALL ON codeinsight.* TO 'fnciuser'@'%';
```

4. Provide the user name and password and the database schema to the person who will install Code Insight.

MySQL Database Settings

Flex Net Code Insight requires the following MySQL database configuration to ensure best performance.



Note • These settings can be edited only by root or the DBA.

Table 1-3 • Required MySQL Database Settings

Property	System Variable	Recommended Value
Storage Engine	default-storage-engine	innodb
Character Set/Collation	character-set-server	UTF-8
InnoDB Buffer Pool Size	innodb_buffer_pool_size	12GB
InnoDB Log File Size	innodb_log_file_size	8GB
Maximum Allowed Packets	max_allowed_packet	100MB



Note • The following settings may only be edit by the root/Administrator user on the database server.

Storage Engine

Select InnoDB as the storage engine. In MySQL 5.7, InnoDB is the default engine, so a change is unlikely to be necessary.

To verify the current storage engine, use the following:

```
SELECT * FROM INFORMATION_SCHEMA.ENGINES;
```

To change the default storage engine, use either of these next procedures.

Linux

As a root user, edit the `my.cnf` file (typically located in `/etc/my.cnf`) by editing (or adding) the following line in the `[mysqld]` section, and then restarting the database server:

```
default-storage-engine=innodb
```

Windows

As an administrator on the system, edit the `my.ini` file (typically located in `C:\ProgramData\MySQL\`) by editing (or adding) the following line in the `[mysqld]` section and restarting the database server:

```
default-storage-engine=innodb
```

Character Set/Collation

Select UTF-8 as the character set when installing the FlexNet Code Insight MySQL database server.

To verify the current character set and collation, use the following commands:

```
SELECT @@character_set_database, @@collation_database;
```

To change the character set, use either of these next procedures.

Linux

As a root user, edit the `my.cnf` file (typically located in `/etc/my.cnf`) by editing (or adding) the following line in the `[mysqld]` section and restarting the database server.

```
character-set-server=utf8
collation-server=utf8_general_ci
```

Windows

As an administrator on the system, edit the `my.ini` file (typically located in `C:\ProgramData\MySQL\`) by editing (or adding) the following line in the `[mysqld]` section and restarting the database server.

```
character-set-server=utf8
collation-server=utf8_general_ci
```

InnoDB Buffer Pool Size

Set the InnoDB buffer pool size to at least 12GB.

To verify the current InnoDB buffer pool setting, use the following command. (The resulting value is in GBs.)

```
SELECT @@innodb_buffer_pool_size/1024/1024/1024;
```

To change the InnoDB buffer pool size, use either of these next procedures.

Linux

As a root user, edit the `my.cnf` file (typically located in `/etc/my.cnf`) by editing (or adding) the following line in the `[mysqld]` section and restarting the database server:

```
innodb_buffer_pool_size=12G
```

Windows

As an administrator on the system, edit the `my.ini` file (typically located in `C:\ProgramData\MySQL\`) by editing (or adding) the following line in the `[mysqld]` section and restarting the database server:

```
[mysqld]
innodb_buffer_pool_size=12G
```

InnoDB Log File Size

Set the InnoDB log file size to at least 8GB.

To verify the current InnoDB log file size, use the following command:

```
show variables like 'innodb_log_file_size';
```

To change the InnoDB log file size, use either of these next procedures.

Linux

As a root user, edit the `my.cnf` file (typically located in `/etc/my.cnf`) by editing (or adding) the following line in the `[mysqld]` section:

```
innodb_log_file_size=8G
```

Restart the database server.

Windows

As an administrator on the system, edit the `my.ini` file (typically located in `C:\ProgramData\MySQL\`) by editing (or adding) the following line in the `[mysqld]` section, and then restart the database server:

```
innodb_log_file_size=8G
```

Maximum Allowed Packets

Set the maximum packet size to 100MB.

To verify the current maximum packet size, use the following command:

```
SHOW VARIABLES LIKE 'max_allowed_packet';
```

To change the maximum packet size, use either of these next procedures.

Linux

As the root user, edit the `my.cnf` file (typically located in `/etc/my.cnf`) by editing (or adding) the following line in the `[mysqld]` section and restarting the database server:

```
max_allowed_packet=100M
```

Windows

As an administrator on the system, edit the `my.ini` file (typically located in `C:\ProgramData\MySQL\`) by editing (or adding) the following line in the `[mysqld]` section and restarting the database server:

```
max_allowed_packet=100M
```

Setting Up the SQL Server Database

Setting up the SQL Server database for Code Insight involves two phases:

- Phase 1: Install the SQL Server Instance
- Phase 2: Set Up the SQL Server Database

The DBA performs these steps.

Phase 1: Install the SQL Server Instance



Task

To install the SQL Server instance:

1. Install the SQL Server instance, following the instructions included with the SQL Server installer. During the installation, select the appropriate options that do the following:
 - Set the character set (or collation) is to `SQL_Latin1_General_CP1_CI_AS`.
 - Enable the SQL Server Agent.

2. When the installation is complete, start up the SQL Server Agent using the instructions provided in the SQL Server documentation. This is a required step for setting up the SQL Server database, described in the next section, [Phase 2: Set Up the SQL Server Database](#).

Phase 2: Set Up the SQL Server Database

Once you have installed the SQL Server instance and have started up the SQL Server Agent, use the following instructions to set up the SQL Server database for Code Insight.



Task

To set up the SQL Server database for Code Insight:

1. Ensure that you have downloaded and extracted the required the Code Insight scripts, as described in [Downloading the Scripts Needed to Set Up the SQL Server Database](#).
2. Understand the purpose of the scripts before executing them:
 - **palamida_serversettings.sql**—This script configures the database server to enable the maximum performance for Code Insight. The script sets the following server parameters:
 - **Cost of parallelism:** 15 (the threshold at which the optimizer chooses parallel processing)
 - **Max degree of parallelism:** Number of threads created specifically for this configuration.
 - **Max memory configuration:** The server's maximum utilization (60 percent) of total memory.
 - **TF:** Trace flags 111, 1118, 2371.

You are strongly recommended to review existing configurations in this script and note their values in case a rollback is needed. However, do not edit this script.

- **palamida_db_creation_with_maintainenceplan.sql**—This script creates the database and schedules maintenance jobs. Specifically, it performs the following operations:
 - Creates a database with 4 data files and 1 log file.
 - Creates a new folder called MSSQLDATA on a non-OS disk. If only one drive exists, the database is created on the OS drive itself.
 - Creates a subfolder with the database name under the MSSQLDATA folder.
 - Creates a daily maintenance job to perform an Update Statistics every 6 hours (no downtime needed).
 - Creates maintenance job to perform an Update Statistics and Index Reorg every two weeks (no downtime needed). The default is to run at 10 pm per server time zone every two weeks.

You can edit some settings in this script as described in Step 4.

3. Ensure that the SQL Server Agent is running.
4. Open the palamida_serversettings.sql script, and execute it.
Do not edit this script.
5. Open the palamida_db_creation_with_maintainenceplan.sql script, edit the @dbname setting if necessary, and then execute the script.

The default value for @dbname is **fncliv7**. To edit this setting, simply overwrite the current value with the preferred database name. If you provide a database name that already exists, the script execution will fail.

6. Create a user who has READ and WRITE permissions on the database (that is, the DBO role). This is the user who will access the Code Insight (SQL Server) database from the Code Insight application.

Network and Firewall Considerations

Configure the servers by specifying a fully qualified domain name (for example, *hostname.domain.com*) or IP address. Enable those port numbers used by FlexNet Code Insight in all of the firewalls. You may use the default port numbers listed below or configure the application to use custom ports.

Table 1-4 • Default Port Numbers Used by FlexNet Code Insight

Port #	Details
3306	MySQL Database Server Access Port
8888/443	Tomcat (http/https)
465	External SMTP (mail) Server
389	External Authentication Directory Server (Active Directory/LDAP)
8005 and 8009	Tomcat Connector and Tomcat Shutdown Ports (local access only)

Setting the Open File Limit for Linux/Unix

The open file limit is a setting that controls the maximum number of open files for a specific user. The default open file limit is typically 1024, but can be set with the `ulimit` command by the root user. For FlexNet Code Insight to function properly in a Linux/Unix environment, the open file limit must be set to handle more than 50k files.



Important • This procedure to increase open file size is absolutely essential for FlexNet Code Insight to function properly on Unix/Linux platforms.

The following are some ways that open file limits are managed, depending on the user's role in the system:

- **soft limit:** set in `/etc/security/limits.conf` by a normal user.
- **hard limit:** set in `/etc/security/limits.conf` by root user.
- **system wide limit:** set in `/etc/sysctl.conf` by root user.

Soft limits are the currently enforced limits, and hard limits are the maximum limits on the system. It is recommended that you log in as the root user so both types of limits may be set accordingly.



Task

To set open file limits on a Linux RedHat system, do the following:

1. In a terminal window, type `ulimit -a` to see a list of current file limits.

2. Locate the *open files (-n)* setting:
 - If the setting is less than 50K, continue to the next step.
 - If the setting is more than 50K, you do not need to perform this procedure.
3. Open the file `/etc/security/limits.conf` and add the following entries:

```
soft nofile 65536
hard nofile 65536
```
4. Save the file and log in again for the changes to take effect.
5. On the command line, type `ulimit -a`, and verify that the *open files (-n)* setting reads 65536.



Note • Other distributions, such as a Ubuntu and CentOS may require a different setting. See instructions for your specific Linux distribution and shell type.

Enabling Secure HTTP Over SSL

To implement SSL, a Secure Site SSL Certificate must exist for each Code Insight Core and Scan server that accepts secure connections. Refer to http://en.wikipedia.org/wiki/HTTP_Secure and <http://tomcat.apache.org/tomcat-8.0-doc/ssl-howto.html> for more details regarding HTTPS.

Use these instructions for enabling an HTTPS connection, including how to procure a certificate:

- [Enabling an HTTPS Connection](#)
- [Purchasing a Secure Site SSL certificate](#)
- [Generating a Self-signed Certificate](#)



Note • For security, we recommend that FlexNet Code Insight always be installed over SSH.

Enabling an HTTPS Connection

Use these instructions to enable the HTTPS connection.



Task **To enable an HTTPS connection, do the following:**

1. Edit the `<CODEINSIGHT_ROOT_DIR>\tomcat\bin\catalina.bat` (or `catalina.sh` depending on your operating system):

```
set -Dcodeinsight.ssl=true (default value is false)
```
2. Back up the `<CODEINSIGHT_ROOT_DIR>\tomcat\conf\server.xml` file to another directory (outside of conf) and then copy the `server.xml` from `<CODEINSIGHT_ROOT_DIR>\tomcat\https` to `<CODEINSIGHT_ROOT_DIR>\tomcat\conf`.

The `server.xml` file contains a default configuration that references a keystore at `<CODEINSIGHT_ROOT_DIR>\tomcat\codeinsight.jks`.

You create or obtain a certificate and save it in a keystore. See step 4 for more information.

3. Search for the text, *FNCI SSL: Edit this section to match your certificate information*. The default values are shown below:

```
<!-- FNCI SSL: Edit this section to match your certificate information -->
<Connector protocol="org.apache.coyote.http11.Http11Protocol"
    port="8888"
    minSpareThreads="25"
    enableLookups="false"
    disableUploadTimeout="true"
    acceptCount="100"
    maxThreads="150"
    maxHttpHeaderSize="8192"
    scheme="https"
    secure="true"
    SSLEnabled="true"
    keystoreFile="codeinsight.jks"
    keystorePass="codeinsight"
    keyAlias="codeinsight"
    keyPass="codeinsight"
    clientAuth="false"
    sslProtocol="TLS"
    ciphers="HIGH: !aNULL: !eNULL: !EXPORT: !DES: !RC4: !MD5: !kRSA"
/>
```

4. Purchase a Secure Site SSL certificate or generate your own self-signed certificate. The procedures for using a purchased certificate and for generating your own differ. Depending upon your type of certificate, consult one of the following sections:

- [Purchasing a Secure Site SSL certificate](#)
- [Generating a Self-signed Certificate](#)

5. Update the `server.xml` file with the following parameters:

keystoreFile: the file name of the keystore containing the certificate
 keystorePass: the password of the keystore
 keyAlias: the alias for the certificate entry in the keystore
 keyPass: the password for the certificate entry



Note • If the keystore and alias passwords are the same, you can specify `keyPass`, `keystorePass` or both.

6. Restart the Tomcat server after making changes to the `server.xml` file or to a keystore. For more information, see [Starting & Stopping Tomcat](#).

Purchasing a Secure Site SSL certificate

The following are two sources for purchasing a Secure Site SSL Certificate:

- <http://www.verisign.com/ssl/buy-ssl-certificates/secure-site-ssl-certificates/index.html>
- <https://www.thawte.com/ssl-digital-certificates/ssl/index.html>

Follow your vendor's instructions for generating a certificate signing request (CSR) and importing the certificate into the keystore.

Creating a Keystore for a Purchased Secure Site SSL Certificate--Example

The following is an example of a command to create a keystore for a Secure Site SSL Certificate on the server:

```
keytool -import -alias "<keyAlias>" -file <yourPurchasedCertificateFile> -keystore  
<CODEINSIGHT_ROOT_DIR>\tomcat\<keystoreFile> -storepass "<keypass>"
```



Task

To use a purchased Secure Site SSL Certificate, you can do the following:

1. Export the certificate and import it into cacerts, which is in <installDirectory>\jre\lib\security.

```
keytool -export -alias "<keyAlias>" -file <file>.crt -keystore <file>.jks  
keytool -delete -alias "<keyAlias>" -keystore cacerts  
keytool -import -alias "<keyAlias>" -keystore cacerts -file <file>.crt
```



Note • The default password for cacerts is changeit.

2. (Optional) To check the contents of the keystore, enter the following command:

```
keytool -list -keystore cacerts shows keystore contents.
```

Generating a Self-signed Certificate



Task

To generate your own self-signed certificate with a keystore in place of a purchased one, do the following:

1. Execute the following command found in the JDK:

```
keytool -genkey -keyalg RSA -sigalg SHA256withRSA -alias "<keyAlias>" -keystore <keystoreFile> -  
storepass "<keypass>" -validity <numDays> -keysize 2048
```

2. Enter the server's host name or IP address when prompted, *What is your first and last name?*

3. Leave the rest of the prompts blank, except for the last one:

Is CN=<yourServerNameOrIPAddress>, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown correct?

For this prompt, type **yes**.

4. Copy the generated keystore to <CODEINSIGHT_ROOT_DIR>\tomcat\.
5. Update the <CODEINSIGHT_ROOT_DIR>\tomcat\conf\server.xml file with the values you provided in the command above so Tomcat can access the generated certificate.

If a self-signed certificate is used on the FlexNet CodeInsight server, each client machine that is used to access FlexNet Code Insight should add a certificate exception to the browser.

Using a Self-signed Certificate--Example

The following example uses a self-signed certificate and codeinsight for keystore, alias and passwords:

1. In catalina.bat, make the following changes:

```
-Dcodeinsight.ssl=true  
tomcat\conf\server.xml replaced by the server.xml in tomcat\https  
cd C:\mywork  
keytool -genkey -keyalg RSA -sigalg SHA256withRSA -alias codeinsight -keystore codeinsight.jks -  
storepass codeinsight -validity 3600 -keysize 2048  
keytool -export -alias codeinsight -file codeinsight.crt -keystore codeinsight.jks  
keytool -delete -alias codeinsight -keystore C:\FlexNetCodeInsight\jre\lib\security\cacerts  
keytool -import -alias codeinsight -keystore C:\FlexNetCodeInsight\jre\lib\security\cacerts -file  
C:\mywork\codeinsight.crt  
keytool -v -list -keystore C:\FlexNetCodeInsight\jre\lib\security\cacerts -alias codeinsight  
copy c:\mywork\codeinsight.jks C:\FlexNetCodeInsight\tomcat\
```

2. Restart Tomcat. For more information, see [Starting & Stopping Tomcat](#).
3. Open a browser and enter **https://<host>:8888/codeinsight**.
4. Navigate to the System Configuration page, and update the scan server configuration.
 - Add a New scan server or select a scan server and edit it.
 - Set the Host name.
 - Set the Port to the https port.



Note • You may need to accept browser warnings the first time that the application comes up; these messages should go away after the initial session is over.

Configuring a Networking Proxy Server Connection

By default, FlexNet Code Insight uses automatic proxy server settings to. However, FlexNet Code Insight can be manually configured to an enterprise proxy with company IT policies.



Task

To manually configure a proxy server connection, do the following:

1. Navigate to the tomcat/bin folder. This folder resides in the directory where FlexNet CodeInsight is installed.
2. Open catalina.bat or catalina.sh for editing.
3. Find line 209 and uncomment it:

```
rem set CATALINA_OPTS=%CATALINA_OPTS% -Dhttps.proxyHost=<HOST> -Dhttps.proxyPort=<PORT> :-  
Dhttps.proxyUser=<USER> -Dhttps.proxyPassword=<PASSWORD>
```
4. Set the following values for the proxy server:
 - **ProxyHost:** IP or Hostname of the proxy.
 - **PorxyPort:** Port being used for proxy.

- **ProxyUser:** Username used to authenticate the proxy. Omit this value for a transparent proxy connection.
 - **ProxyPassword:** Password used to authenticate the proxy. Omit this value for a transparent proxy connection.
5. Restart the Tomcat server so the proxy server changes take effect. For information about restarting Tomcat, see [Starting & Stopping Tomcat](#).

Installing FlexNet Code Insight

Use the following instructions to install FlexNet Code Insight:

- [Gathering the Required Files](#)
- [Launching the Installer](#)

Gathering the Required Files

While installing FlexNet Code Insight, you will be asked to provide your license key and directory paths for files that are needed for the application to function. In addition, you will choose the type of installation to perform. The following is a list of the items and information to collect before beginning the installation:

- The license key file, `codeinsight.key`. If you do not have a license key file, visit the Flexera Customer Community at <https://flexeracomunity.force.com/customer/CCContactSupport>.
- The appropriate JDBC driver connector file for the database:
 - **For MySQL:** The connector file `mysql-connector-java-5.1.41-bin.jar`. If you do not have a connector file, download one from the Oracle MySQL webpage: <https://dev.mysql.com/downloads/connector/j/>.
 - **For SQL Server:** The connector file `mssql-jdbc-6.4.0.jre8.jar`. If you do not have a connector file, download one from the Microsoft webpage: <https://www.microsoft.com/en-us/download/details.aspx?id=56615>.
- The type of installation you will perform:
 - **Standalone:** Configure your computer as both the core and scan server. This is the recommended configuration.
 - **Core:** Configure your computer as the core server.
 - **Scanner:** Configure your computer as the scan server.

The Core Server controls your Web UI Client. The Scan Server is where actual scanning is performed.

Additionally, ensure that you have met the prerequisites listed in [System Requirements](#).

Launching the Installer

After you create a database with remote access privileges, you can use the Installer to install FlexNet Code Insight in a Windows or Linux environment.



Note • You can cancel the installation by clicking **Cancel** on any installation panel.

**Task**

To install FlexNet Code Insight, do the following:

1. Follow the installation steps for your environment:

On Windows

Download the Windows installer (FlexNetCodeInsight.exe), and then navigate to the directory where you downloaded the file. Double-click the filename and follow the prompts to install FlexNet Code Insight in a Windows environment.

On Linux

Download the Linux installer file (FlexNetCodeInsight.bin), and then navigate to the directory where you downloaded the file. Launch FlexNetCodeInsight.bin and follow the prompts to install FlexNet Code Insight in a Linux environment.

2. When the installation is complete, do the following:
 - a. Start the Tomcat server if it is not already running. See [Starting & Stopping Tomcat](#).
 - b. Launch Code Insight by following the procedures in [Opening FlexNet Code Insight](#).



Important • If the installation does not complete, contact <https://flexeracommunity.force.com/customer/CCContactSupport>.

Running FlexNet Code Insight as a Service

Running FlexNet Code Insight as a service whenever your system starts up can save time. This section provides the procedure to configure FlexNet Code Insight as a service in both a Windows environment and a Linux (RedHat 7, CentOS 7) environment:

- [In a Windows Environment](#)
- [In a Linux Environment](#)

In a Windows Environment

Perform the following procedure to run FlexNet Code Insight as a Windows service.

**Task**

To run FlexNet Code Insight as a Windows service, do the following:

1. Using the command prompt, navigate to:
`<CODE_INSIGHT_ROOT_DIR>\tomcat\bin.`
2. Stop the Tomcat server. See [Starting & Stopping Tomcat](#).
3. Open the service.bat file with a text editor.

4. Under the `Set default Service name` comment, set the following parameters:
 - `SERVICE_NAME=CodeInsight`
 - `DISPLAYNAME=FlexNet Code Insight`
5. Change the Description to reflect the name of the service, which is *Code Insight*.
6. On the `JvmOptions` line, add the following to the list:
 - `-Dcodeinsight.ssl=false`
 - `-DcodeinsightInstallPath=<codeinsightrootdirectory>`

The `codeinsightrootdirectory` is the directory path where FlexNet Code Insight is installed.



Note • Remember to separate the `JvmOptions` entries with a semi-colon (;).

7. Change the `JvmMs` initial memory setting to 8192m. The default entry is 128.
8. Change the `JvmMx` maximum memory setting to 16384m. The default entry is 256.
9. Save the `service.bat` file and exit the text editor.
10. Execute the `service.bat install` command to install the Apache Tomcat Windows service.
11. When the service is installed, open **Windows Services** and search for the Service name you specified in step 4. In this case, it is *CodeInsight*.
12. Right click on the *CodeInsight* service and select **Start**.

In a Linux Environment

Perform the following procedure to run FlexNet Code Insight as a service on Linux (RedHat 7 or CentOS 7).



Task

To run FlexNet Code Insight as a service in Linux, do the following:

1. Create a file named `OpenSpecimen.service` with the following contents. (Note that this file name is case-sensitive when referenced in commands used in this procedure.)


```
[Unit]
Description=Tomcat Service OpenSpecimen.service
After=syslog.target network.target
[Service]
Type=forking
ExecStart=/install path of tomcat/bin/startup.sh
#Eg. ExecStart=/home/qaadmin/FlexNetCodeInsight/tomcat/bin/startup.shsh
ExecStop=/bin/kill -15 $MAINPID
[Install]
WantedBy=multi-user.target
```
2. Copy the `OpenSpecimen.service` file to the `/etc/systemd/system` directory.
3. Stop the Tomcat server. See [Starting & Stopping Tomcat](#).

4. Execute the following command to notify systemd that the OpenSpecimen service has been added:

```
$ sudo systemctl daemon-reload
```

5. Use the following commands to start, stop, or restart the OpenSpecimen service:

```
$ sudo systemctl start OpenSpecimen.service
$ sudo systemctl stop OpenSpecimen.service
$ sudo systemctl restart OpenSpecimen.service
```

6. Execute the following command to enable the starting of OpenSpecimen upon booting:

```
systemctl enable OpenSpecimen.service
```

From this point on, when you start your system, FlexNet Code Insight will start up automatically.

Starting & Stopping Tomcat

From time to time, it is necessary to start and stop the Tomcat server. For example, if this is the first time you have installed FlexNet Code Insight, or if you have recently upgraded FlexNet Code Insight or shut down your Tomcat server, you must restart it before you can connect to FlexNet Code Insight in a browser.



Task

To start the Tomcat server, do the following:

1. Navigate to the directory where FlexNet Code Insight is installed and open the tomcat\bin directory. For example, C:\FlexNetCodeInsight\tomcat\bin.
2. Execute the startup.bat file for Windows or the startup.sh file for Linux. As the Tomcat startup runs, messages are displayed on the Tomcat console. The Tomcat startup may take several minutes to complete. When a startup message similar to the following appears in the Tomcat console, you can open FlexNet Code Insight in your browser:

```
10-Aug-2017 10:06:34.796 INFO [main] org.apache.catalina.startup.Catalina.start Server startup in 58823 ms
```



Task

To shut down the Tomcat server, do the following:

1. Navigate to the directory where FlexNet Code Insight is installed and open the tomcat\bin directory. For example, C:\FlexNetCodeInsight\tomcat\bin.
2. Execute the shutdown.bat file for Windows or the shutdown.sh file for Linux.

Opening FlexNet Code Insight

FlexNet Code Insight runs in your web browser. This section explains how to start FlexNet Code Insight and access the **Dashboard**.



Task

To open FlexNet Code Insight, do the following:

1. Launch a web browser and navigate to the following URL, entering the server host name provided by your FlexNet Code Insight administrator:

`http://<your_server_host_name>:PORTNUMBER/codeinsight/`

For example, `http://localhost:8888/codeinsight/`.

The FlexNet Code Insight Login page opens.



Note • If you are unsure about your server host name, contact your system administrator for guidance.

2. Enter your **username** and **password**.



Note • The default login name is **admin**; the default password is **Password123**. Your installation may require a different login name and password. If you are unsure about what to enter, contact your system administrator for guidance.

3. Click **Login**. The **FlexNet Code Insight Dashboard** appears.

Uninstalling FlexNet Code Insight

An uninstaller for FlexNet Code Insight is available in the directory where the product is installed. The following procedures show you how to uninstall FlexNet Code Insight in a Windows and a Linux environment. Instructions are also provided to drop the SQL Server database used as the Code Insight database, should this action be necessary.

- [Uninstalling on Windows](#)
- [Uninstalling on Linux](#)
- [Dropping the SQL Server Database](#)

Uninstalling on Windows

Use the following procedure to uninstall Code Insight on a Windows machine.



Task

To uninstall FlexNet Code Insight in Windows:

1. Navigate to the directory where FlexNet Code Insight is installed.
2. Open the **Uninstall_FlexNetCodeInsight** folder.
3. Double-click **Uninstall FlexNetCodeInsight.exe**.
4. Follow the on-screen prompts to uninstall FlexNet Code Insight. The uninstall process will leave behind some files. Review them and delete as needed.

Uninstalling on Linux

Use the following procedure to uninstall Code Insight on a Linux machine.



Task

To uninstall FlexNet Code Insight in Linux:

1. Navigate to the directory where FlexNet Code Insight is installed.
2. Open the **Uninstall_FlexNetCodeInsight** folder.
3. Execute **Uninstall FlexNetCodeInsight** command and follow the on-screen prompts to uninstall FlexNet Code Insight. The uninstall process will leave behind some files. Review them and delete as needed.

Dropping the SQL Server Database

If you need to drop the SQL Server database used as the Code Insight database, follow this procedure. Dropping the database also drops its maintenance plans.



Task

To drop the SQL Server database and its maintenance plans:

1. If you have not already done so, download the `palamida_db_drop_with_maintainenceplan.sql` script. See [Downloading the Scripts Needed to Set Up the SQL Server Database](#).
2. Open the script, and set the `@dbname` value to the name of the database to be dropped (if the value is not set to the correct name).
3. Execute the script.

Contacting Support

If you need further support, please submit your questions through our online **Customer Community** portal:

<https://flexeracommunity.force.com/customer/>

If you do not have a login to the Customer Community, you can request one on the **Login Request** page of our site:

<https://flexeracommunity.force.com/customer/CCLoginRequest>

If you are unable to use the steps above, please visit the following site for other options to reach out to Flexera Support:

<https://flexeracommunity.force.com/customer/CCContactSupport>

Configuring FlexNet Code Insight

After FlexNet Code Insight had been installed, the administrator must perform a number of configuration tasks before the user can begin using Code Insight. This section describes these configuration tasks:

- [Creating or Editing a Scan Server](#)
- [Managing Users](#)
- [Setting the Electronic Update Frequency](#)
- [Configuring an Email Server](#)
- [Configuring LDAP](#)
- [Configuring FlexNet Code Insight to Use Single Sign-On](#)
- [Managing Scan Profiles](#)
- [Setting Project Defaults](#)
- [About FlexNet Code Insight Server REST APIs](#)



Note • The first time you open FlexNet Code Insight, an electronic update will begin. It may take 2 to 4 hours for the electronic update to complete. You cannot use the application to scan files until the update finishes. However, you can configure FlexNet Code Insight while the update is in progress.

Additionally, see [About FlexNet Code Insight Server REST APIs](#) in this chapter for information about Code Insight REST APIs that enable you to create your administrative tool for managing scan operations and retrieving data from scan results.

Creating or Editing a Scan Server

The Scan Server scans the source code and binary files that make up your codebases to help you identify open source code that may expose your applications to compliance issues and security vulnerabilities. You must set up a scan server before scanning code.



Note • *FlexNet Code Insight supports only one scan-server configuration.*

**Task**

To create or edit your scan server, do the following:

1. On the **FlexNet Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs.
2. Select the **Scan Servers** tab.
3. If you do not have a scan server configured, click **New**; or to edit your already-defined scan server, select it from the **Scan Servers** drop-down list, and click **Edit**. The **Scan Server** dialog appears.
4. Complete or update the fields on the dialog:
 - **Alias**: The name for the scan server.
 - **Host**: The IP address of the host computer for the scan server. If the scan server is on the same machine as the core server, enter *localhost*.
 - **Port**: The host port of the scan server. By default, the port is 8888.
 - **CL Path (Optional)**: The path for the FlexNet Code Insight Compliance Library, which is provided on an USB SSD drive. However, using FlexNet Code Insight's automated discovery, you can to perform a scan even before obtaining the Compliance Library or setting up a scan server. For more information, see the *FlexNet Code Insight User Guide*.
 - **Codebase Path**: The path on the scan server where FlexNet Code Insight will store and manage all uploaded code. You should have adequate disk space to store the codebases. Recommended starting size for this directory is 500GB.



Note • *If you are unsure about what to enter in any of these fields, contact SCA Support for guidance.*

Managing Users

The following topics describe how to manage FlexNet Code Insight users:

- [Creating or Editing Users](#)
- [Finding Users](#)
- [Disabling User Accounts](#)

Creating or Editing Users

The following procedure describes how to create or edit users for your FlexNet Code Insight installation. (If you are using an LDAP server to sync the user data, you can skip this procedure. To configure an LDAP server, see [Configuring LDAP](#).)




Note • The first time you open FlexNet Code Insight, an electronic update will begin. It may take 2 to 4 hours for the electronic update to complete. You cannot use the application to scan files until the update finishes. However, you can configure FlexNet Code Insight users while the update is in progress.



Task

To create a user, do the following:

1. On the **FlexNet Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs.
2. Select the **Users** tab, which lists all current users.
3. To create a new user, click **Add User**; or to edit an existing user, click the Edit icon  .
The **Add User** or **Edit User** dialog appears.
4. Enter information in the fields to create or edit the user:
 - **Login**: The user's login name.
 - **First Name**: The user's first name.
 - **Last Name**: The user's last name.
 - **Email**: The user's email address.
 - **Password**: The user's password, which should be a minimum of 8 characters with no spaces and have at least one number and one capital letter.
 - **Password Confirm**: Reenter the password from the field above.
 - **Question**: A security question that can be answered by the user to retrieve a lost password. The question must be a minimum of 3 characters.
 - **Answer**: The answer to the security question.
 - **Permissions**: Choose one or both of the following permissions:
 - **Administrator**: Provides permission to manage users and application configuration settings for FlexNet Code Insight.
 - **Policy Management**: Provides permission to manage policies in FlexNet Code Insight.
5. When you finish entering information for the user, click **Submit**. The **Success** dialog appears, telling you that the user has been saved.
6. Click **OK**. If you created a user, the user will appear in the list.

Finding Users

As a system administrator or project owner, you might need to find FlexNet Code Insight users to manage their permissions. You can search for users on the **Users** tab or on the **Project Summary** page.



Task

To find users, do the following:

1. On the **FlexNet Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs.
2. Select the **Users** tab.
3. In the **Enter Search Criteria** field, enter a character string by which to search user information in any of the fields.
4. Click **Search**.

Disabling User Accounts

FlexNet Code Insight supports disabling user accounts in the browser.



Note • The Admin user account is created automatically; it cannot be disabled.



Task

To disable user accounts, do the following:

1. On the **FlexNet Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs.
2. Select the **Users** tab.
3. Click the **Edit** icon (✎) in the **Actions** column for the user account you want to disable. The **Edit User** dialog appears.
4. Select the **Disable Account** checkbox, and click **Submit**. The **Success** dialog appears.
5. Click **OK**. The user account is now disabled. The user will receive the message, “Invalid Username and/or Password. If you believe you entered a valid user, please contact your System Administrator” when attempting to log into FlexNet Code Insight.

Setting the Electronic Update Frequency

Frequent updates enable you to receive the latest vulnerability or other component information as quickly as it is available. However, scans cannot be performed during the update process, but a scan process that is already underway will not be interrupted when the update process is triggered.

The default electronic update to product data is daily at 1 am, but FlexNet Code Insight provides the ability to manage the frequency with which product data is updated. The following procedure explains how to set the update frequency.



Task

To set the update frequency, do the following:

1. On the **FlexNet Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs.
2. Select the **Electronic Updates** tab.

3. Select a frequency from the **Update Frequency** pulldown:
 - **Never:** If you select **Never**, the other frequency fields disappear.
 - **Daily:** If you select **Daily**, you must select a time from the **Time** pulldown.
 - **Weekly:** If you select **Weekly**, you must select a time from the **Time** pulldown and a day from the **Select a day** pulldown.
4. If you selected a **Daily** or **Weekly** frequency, select a time from the **Select a time** pulldown.
5. If you selected a **Weekly** frequency, select a day from the **Select a day** pulldown.
6. When you have finished setting the update frequency, select **Save**. After the update frequency has been updated, the **Success** prompt appears.
7. Click **OK** to return to the **Admin selection** page.



Note • If you click **Schedule Update**, the update will take place immediately. The message, “An electronic update is currently being processed... please wait...”, appears in the **Electronic Update** field.

Configuring an Email Server

FlexNet Code Insight can send email alerts that are triggered by certain events. For example, when a scan completes or when a new vulnerability is detected in the project inventory. It is highly recommended that the email server configuration be set up for the application. Email server configuration is available in FlexNet Code Insight in the Administration tabs. This section provides the procedure for configuring email.



Task

To configure your email server, do the following:

1. On the **FlexNet Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs.
2. Select the **Email Server** tab.
3. Enter information and make selections in the fields:
 - **Enable Email Server:** Select **Yes** to enable FlexNet Code Insight to use the email server or **No** to leave it disabled. The default is **No**. The rest of the fields on this page are not available until you select **Yes**.
 - **Sender's Email Address:** Enter the email address of the sender.
 - **SMTP Host Name:** Enter the SMTP host name.
 - **SMTP Host Port:** Enter the port number of the SMTP host.
 - **SMTP User Name:** Enter the SMTP user name. This field can be left blank for anonymous SMTP configuration.
 - **SMTP User Password:** Enter the SMTP user password. This field can be left blank for anonymous SMTP configuration.
 - **Enable SMTP over TLS:** Select **Yes** to use Transport Layer Security (TLS) to secure email over SMTP or select **No** to leave this option disabled.

4. Click **Save** to save your settings.

Configuring LDAP

The LDAP option allows you to use an LDAP server to import user name data into FlexNet Code Insight and for authentication, as described in the following topics:

- [Synchronizing User Name Data](#)
- [Setting Up a User Search Filter](#)
- [Sample Search Query](#)
- [Server Paging](#)
- [User Authentication](#)
- [LDAP over SSL](#)
- [Implementing LDAP](#)

Synchronizing User Name Data

FlexNet Code Insight provides the ability to import user name data from LDAP. This section explains the type of user name data that is imported.

User Metadata

The metadata for each user (name, email, etc.) is pulled from LDAP and refreshed in the FlexNet Code Insight database at a regular frequency via a scheduler module running within FlexNet Code Insight. The data synchronization is a one-way pull from LDAP into the FlexNet Code Insight database. This action overwrites the existing data in the FlexNet Code Insight database. User data for those users that do not exist in LDAP is not affected by this process.

Disabled Users

Users who are disabled in FlexNet Code Insight will still have their data synchronized with LDAP, but will have the disabled flag set to “true” and will not be granted access to the application.

Setting Up a User Search Filter

To pull only the required users into FlexNet Code Insight, it is important to configure the Search Base and Search Query entries, which appear on the **LDAP tab** of the FlexNet Code Insight user interface, properly. The Search Base is typically the root node under which you can store all the desired users. The Search Query allows LDAP queries based on user attributes. We recommend creating a FlexNet Code Insight system-specific group and making all of the desired users part of this group.

Sample Search Query

LDAP search queries can be entered in the **LDAP Search Query** field on the **LDAP** tab. For example, the following query pulls only desired users into FlexNet Code Insight:

```
(&(objectClass=person)(memberOf=CN=Code InsightGroup,CN=Users,DC=ad,DC=Code Insight,DC=com))
```

Server Paging

LDAP and Active Directory support server paging controls the number of records the system is pulling at any given time. Configure the LDAP Page Size entries as desired. The default page size is 1000.



Note • SunOne Directory Server does not support server paging in certain releases <http://kb.globalscape.com/KnowledgebaseArticle10218.aspx>. If you are using SunOne Directory Server, ensure that server paging is disabled.

User Authentication

You can use an existing LDAP server to verify users when they log into FlexNet Code Insight. FlexNet Code Insight does not store LDAP passwords. All authentication happens on the LDAP server. After an LDAP user enters a username and password, the credentials are sent to the LDAP instance. If LDAP confirms that the user is valid, FlexNet Code Insight grants access.



Note • If you configure LDAP to provide login security, the built-in FlexNet Code Insight login security will not be used.

LDAP over SSL

SSL provides data encryption security for user information passed over the network. You must use ldaps://URL with 636 port, which is the default dedicated port for SSL.

Implementing LDAP

This section explains the basic procedure for implementing LDAP in FlexNet Code Insight. For detailed descriptions of the fields on the LDAP tab, see the “LDAP tab” topic in the online help or *FlexNet Code Insight User Guide*.



Task

To implement LDAP, do the following:

1. On the **FlexNet Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs.
2. Select the **LDAP** tab.
3. Select **Yes** in the **Enable LDAP** field and complete the rest of the fields on the **LDAP** tab. See “LDAP Tab” in the online help or in the *FlexNet Code Insight User Guide* for descriptions of all the fields on the **LDAP** tab.

4. (Optional) Select **Test LDAP Server Connection** to ensure that FlexNet Code Insight is properly connected to the LDAP server. The connection will be tested with the values displayed in the fields on the **LDAP** tab.
5. When you finish entering information in the fields, select **Save** to save your changes to the LDAP configuration.
6. (Optional) Select **Sync Now** to save your settings and synchronize them with the user data on the LDAP server. If you do not select **Sync Now**, the user synchronization will be done at the time specified in the **LDAP User Sync Frequency** field.

Configuring FlexNet Code Insight to Use Single Sign-On

Single sign-on (SSO) is an authentication service that enables a user to use one set of credentials (usually a name and password) to access multiple applications. This service involves an exchange of SAML (Security Assertion Markup Language) protocol messages between the user, the identity provider, and the service provider.

The Identity Provider (also called an IdP) is any SSO service, such as Okta, Ping Federate, and others, offering SAML authentication services. The Service Provider (also called an SP) is an application, such as FlexNet Code Insight, that is configured to participate in the SSO service. When a Service Provider user logs in using credentials for an SSO session, a SAML message is sent to the Identity Provider, requesting user authentication. If the user password is valid, the Identity Provider returns a SAML message, stating that the user is logged in at the Identity Provider. The user, in turn, is logged into the Service Provider.

The FlexNet Code Insight administrator can use the instructions in these sections to configure Code Insight as a Service Provider in an SSO session:

- [Prerequisite Tasks for Configuring Code Insight for SSO](#)
- [Configuring Code Insight for SSO](#)
- [Log In Using SSO Credentials](#)
- [Example Okta Setup for Code Insight SSO](#)

Prerequisite Tasks for Configuring Code Insight for SSO

Perform the following tasks before configuring Code Insight for SSO:

- [Configure HTTPS on the FlexNet Code Insight Server](#)
- [Set Up SSO Users](#)

Configure HTTPS on the FlexNet Code Insight Server

The HTTPS communication protocol must be used to exchange SAML messages between the SP and IdP. For instructions on configuring HTTPS on the Code Insight server, see [Enabling Secure HTTP Over SSL](#) in the “Installing FlexNet Code Insight” chapter.

The keystore that you use to configure HTTPS can be used for SSO configuration. Alternatively, you can create a separate keystore for SSO, using the same instructions found in [Enabling Secure HTTP Over SSL](#).

Set Up SSO Users

You can define SSO users for Code Insight with or without LDAP.

With LDAP

If you intend for SSO to integrate with your LDAP server for user access to Code Insight, follow these rules:

- Make sure that Code Insight and the Service Provider are configured for the LDAP server. For instructions to configure Code Insight, see [Configuring LDAP](#).

To configure the Service Provider, follow the Service Provider instructions.

- When setting up users on the LDAP server, ensure that the user's login is the user's email address.
- Synchronize users from the LDAP server to the Identity Provider first, using the Identity Provider's instructions. Then synchronize the users from the LDAP server to Code Insight. See [Configuring LDAP](#).

Without LDAP

If you do not use LDAP, you must manually create the SSO users both in FlexNet Code Insight (see [Managing Users](#)) and at the Identity Provider site, ensuring that the user information is the same in both locations.

Ensure that the user's login is the user's email address.

Configuring Code Insight for SSO

Follow these steps for configuring Code Insight for SSO:

- [Step 1: Copy the Directory That Will Contain Provider Metadata](#)
- [Step 2: Prepare the Environment Properties File](#)
- [Step 3: Configure the SSO Common Properties File](#)
- [Step 4: Customize the Sample Service Provider Metadata File](#)
- [Step 5: Obtain the Identity Provider Metadata File](#)

Note that, in these instructions, `SCA_install_home` refers to the Code Insight installation location.

Step 1: Copy the Directory That Will Contain Provider Metadata

Copy the security directory from `SCA_install_home/samples/sso/config/core` to `SCA_install_home/config/core`.

This directory will serve as the storage location for the Service Provider and Identity Provider metadata files, as described in [Step 5: Obtain the Identity Provider Metadata File](#) and [Step 4: Customize the Sample Service Provider Metadata File](#).

Step 2: Prepare the Environment Properties File

This step prepares the `env.properties` file to enable SSO on the Code Insight server.



Task

To prepare the “env.properties” file:

1. Copy the env.properties file from `SCA_install_home/samples/sso/config` to `SCA_install_home/config/core`.
2. In a text editor, open the `SCA_install_home/config/core/env.properties` file, and ensure that the value of the following property to `sso`.

`spring.profiles.active=sso`
3. Save the file.

Step 3: Configure the SSO Common Properties File

This step configures the `core.sso.common.properties` file to enable SSO on the Code Insight server.



Task

To prepare the “core.sso.common.properties” file:

1. Copy the `core.sso.common.properties` file from `SCA_install_home/samples/sso/config` to `SCA_install_home/config/core`.
2. In a text editor, open the `SCA_install_home/config/core/core.sso.common.properties` file. The following shows the file contents:

```
## this file contains all sso placeholder values.
saml.keystore=file:///c:/<path>/keystore.jks
saml.keystore.password=keystore_password
saml.keystore.alias=keystore_alias
saml.keystore.alias.password=keystore_alias_password

# for extendedMetadata configuration
saml.metadata.local=true
saml.metadata.alias=
saml.metadata.idpDiscoveryEnabled=false
saml.metadata.idpDiscoveryURL=
saml.metadata.idpDiscoveryResponseURL=
saml.metadata.ecpEnabled=false
saml.metadata.securityProfile=metaiop
saml.metadata.sslSecurityProfile=pkix
saml.metadata.sslHostnameVerification=default
saml.metadata.signingKey=keystore_alias
saml.metadata.signingAlgorithm=null
saml.metadata.signMetadata=false
saml.metadata.encryptionKey=keystore_alias
saml.metadata.tlsKey=
#private Set<String> trustedKeys=
saml.metadata.requireLogoutRequestSigned=false
saml.metadata.requireLogoutResponseSigned=false
saml.metadata.requireArtifactResolveSigned=false
saml.metadata.supportUnsolicitedResponse=true
#for SP
saml.entity.id=ww:xx:yy:zz
saml.base.url=https://myhost.mycompany.com:8443
```


- Update the properties (highlighted above) required for Service Provider security and identification, and then save the file. The properties that you need to edit or that require explicit configuration are described in this table:

SSO Property	Description
saml.keystore	Enter the path and name of the keystore that you created for SSO. This can be the same keystore that you are using for HTTPS or a different one. See Configure HTTPS on the FlexNet Code Insight Server in the “Installing FlexNet Code Insight” chapter for more information.
saml.keystore.password	Enter the password for the keystore.
saml.keystore.alias	Enter the alias defined for the private key contained in the keystore.
saml.keystore.alias.password	Enter the password for the private key alias.
saml.metadata.alias	Provide your metadata alias, if one exists; or leave this field blank (or enter defaultAlias) to use the default metadata alias.
saml.metadata.idpDiscoveryURL	Leave this field blank. Do not enter null .
saml.metadata.idpDiscoveryResponseURL	Leave this field blank. Do not enter null .
saml.metadata.signingKey	Enter the path and name of the keystore you created for SSO. (This is the same value entered for the <code>saml.keystore</code> property.)
saml.metadata.encryptionKey	Enter the path and name of the keystore you created for SSO. (This is the same value entered for the <code>saml.keystore</code> property.)
saml.metadata.tlsKey	Enter the alias of private key generated for SSL/TLS client authentication, if one exists; or leave this field blank to use the default TLS key alias.
saml.entity.id	<p>Enter a unique identifier for your Code Insight server as a Service Provider. The recommended value is the hostname for the Code Insight server.</p> <p>Note that, even though the server’s hostname is the recommended value, the entity ID is an immutable value identifying the Service Provider in an SSO session; it is not used to identify a location.</p>
saml.base.url	The HTTPS URL handling the Service Provider’s user sign-in requests. This is usually the URL for your Code Insight server in <code>HTTPS://myhost.mycompany.com:port</code> format. Note that the default port for the Code Insight server is 8443.

Step 4: Customize the Sample Service Provider Metadata File

This step customizes the sample Service Provider metadata file for your Code Insight server.



Task *To customize the sample Service Provider metadata file:*

1. In a text editor, open the `SCA_install_home/config/core/security/SPMetadata.xml` file.
2. Update the following properties, and save the file:

SSO Property	Description
<code>entityID="ENTITY_VALUE"</code>	Replace ENTITY_VALUE with the same entity ID as the one you provided the <code>env.properties</code> file in Step 2: Prepare the Environment Properties File .
<code>SingleLogoutService...FULLY_QUALIFIEDHOSTNAME...</code>	Replace FULLY_QUALIFIEDHOSTNAME with the fully qualified hostname of the Code Insight server.
<code>AssertionConsumerService...FULLY_QUALIFIEDHOSTNAME...</code>	Replace FULLY_QUALIFIEDHOSTNAME with the fully qualified hostname of the Code Insight server.

Step 5: Obtain the Identity Provider Metadata File

This final step in setting up SSO for Code Insight is to obtain the Identity Provider metadata file. The Identity Provider might require that you send the Code Insight `SPMetadata.xml` file (set up in [Step 4: Customize the Sample Service Provider Metadata File](#)) in order to provide the Identity Provider metadata file.

Alternatively, you might be required to generate the Identity Provider metadata file using the Identity Provider UI. You will need to provide the single-sign-on URL for Code Insight (also specified in the `SPMetadata.xml`):

`https://myhost.mycompany.com:8443/codeinsight/saml/SSO`



Task *To obtain the Identity Provider metadata:*

1. Follow the Identity Provider's instructions for obtaining the Identity Provider metadata.
2. Once you obtain the Identity Provider metadata, save it as `IDPMetadata.xml` in the `SCA_install_home/config/core/security` directory.

Log In Using SSO Credentials

Once you complete the steps described in this section, Code Insight users defined as SSO users should be able to log in to an SSO session managed by the Identity Provider and obtain access to Code Insight.

Example Okta Setup for Code Insight SSO

Code Insight provides an example walk-through for using Okta to set up Code Insight for SSO. To obtain these instructions, download the `SSO_With_Okta.pdf` from the Flexera Customer Community site:

<https://flexeracommunity.force.com/customer/>

Managing Scan Profiles

The following topics describe how to manage scan profiles:

- [Creating or Editing Scan Profiles](#)
- [Scan Profile Fields](#)
- [About Scanning without the Compliance Library](#)
- [Creating Exclusion Patterns for Scan Profiles](#)

Creating or Editing Scan Profiles

A scan profile is a set of predefined scan settings that are grouped together that can be applied at scan time. By default, three scan profiles are provided:

- Basic Scan Profile (without CL)
- Standard Scan Profile
- Comprehensive Scan Profile

The table in the [Scan Profile Fields](#) section identifies the scan functions that are included with each scan type.

In most cases, the pre-defined scan profiles are enough to get started. However, if they do not meet your needs, you can create your own custom scan profiles. When a scan profile is created, the data from the Standard Scan Profile is copied, including any search terms and exclusions. However, you can update any of this information the scan profile you are creating.

You can also edit information in existing scan profiles (except the Standard Scan Profile).



Note • Scan profiles changes do not affect the current scan. Changes are applied to the next scheduled scan.



Task

To create or edit a new scan profile, do the following:

1. On the **FlexNet Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs.
2. Select the **Scan Profiles** tab.
3. Click **New** or **Edit** next to the drop-down field listing the existing scan profiles. The **Create** (or **Edit**) **Scan Profile** dialog appears.
4. Complete the fields on the dialog. See the next section, [Scan Profile Fields](#).
5. Click **Save** to save the scan profile.

Scan Profile Fields

The following table summarizes the function of each field in the scan profile. It also notes which fields are valid for default scan profiles shipped with Code Insight:

Table 2-1 • Scan Field Descriptions and Default Scan Profile Support

Field	Description	Basic	Standard	Comprehensive
Perform Package/License Discovery in Archives	Select this option to have the scanner recursively perform package discovery and license detection within all archive files encountered in the project codebase. By default, this option is unselected.	X	X	X
Dependency Support	<p>Determine the level of dependency scanning to be performed by the scanner. The available options include:</p> <ul style="list-style-type: none"> • No Dependencies: Only top-level inventory items are reported without any dependencies. (Default) • Only First Level Dependencies: Only first-level (or direct) dependencies are reported along with top-level inventory items. • All Transitive Dependencies: All first-level and transitive dependencies are reported along with top-level inventory items. The scanner calls out to the relevant package management repository to obtain transitive dependency information. <p>This option is supported only for Java/Maven through <code>pom.xml</code> files and NPM through <code>package.json</code> files. Additional technologies will be supported in future releases.</p>	X	X	X
Automatically Add Related Files to Inventory	Select this option to have the system associate additional files to existing inventory items based on the data available in automatic detection rules. The automatic file mappings are marked with either high or low confidence.	X	X	X
Exact Matches	Select this option to have the scanner record exact matches for scanned files based on data from the Compliance Library (CL).		X	X
Source Code Matches	Select this option to have the scanner record source code matches for scanned files based on data from the Compliance Library (CL).			X

Table 2-1 • Scan Field Descriptions and Default Scan Profile Support (cont.)

Field	Description	Basic	Standard	Comprehensive
Include System Identified Files	(Available only when Source Code Matches is selected) Select this option if you want the scanner <i>not</i> to perform source code matching for files that are already associated with one or more inventory items.			X
Include Files with Exact Matches	(Available only when Source Code Matches is selected) Select this option if you want the scanner <i>not</i> to perform source code matching for files that have exact matches.			X
Search Terms	Provide a list of search terms to be used in the scan.	X	X	X
Scan Exclusions	Provide a list of file extensions to be excluded from the scan. Also see Creating Exclusion Patterns for Scan Profiles .	X	X	X



Note • Comprehensive and Standard Scan Profiles rely on data stored in the Compliance Library (CL) to detect evidence for Exact Matches and Source Code Matches.

About Scanning without the Compliance Library

By default, when FlexNet Code Insight scans a codebase, it uses the data in the Compliance Library (CL) to provide evidence for Exact matches and Source code fingerprint matches. The Compliance Library, which is over 500 GB, is provided on a hard drive, which should be connected to the server where you have installed FlexNet Code Insight.

However, if you do not have access to the Compliance Library, such as when running FlexNet Code Insight on a virtual machine, you can still perform a basic scan (using the Basic Scan Profile) on your codebase that will generate inventory and detect vulnerabilities, find evidence based on pre-defined search terms, emails, and URLs, as well as employ all automated detection techniques. In the absence of a Compliance Library, FlexNet Code Insight will not detect Exact matches and Source code fingerprint matches.

You can also create a custom basic scan profile with your own pre-defined search terms, as well as specify scan exclusions for folders or files to exclude from the codebase scan, such as `**/.git` or `**/.hg`.

Creating Exclusion Patterns for Scan Profiles

Flex Net Code Insight provides the ability to create exclusion patterns for use in your scans and to add them to your scan profile in **Create** (or **Edit**) **Scan Profile** page. This section provides information about the syntax required when creating exclusion patterns and examples of valid exclusion patterns.

Flex Net Code Insight uses Apache Ant path-style syntax to exclude files during scanning. Patterns are paths that are relative to a base directory. Only files found in or below the base directory are considered for exclusion. For in-depth information about *ant* exclusion patterns, see <https://ant.apache.org/manual/dirtasks.html>.



Note • Exclusion patterns are not validated.

Using the Single Asterisk (*) and Question Mark (?)

Using a single asterisk (*) matches zero or more characters. Using the question mark (?) matches one character. If you create an exclusion pattern of *.xml, and add it to the list of Scan Exclusions in FNCI, your scan will exclude files such as x.xml, FooBar.xml, codeinsight.xml but not codeinsight.jar because it does not end with .xml.

If you create an exclusion pattern of ?.codeinsight and add it to your list of Scan Exclusions in FNCI, your scan will exclude files such as x.codeinsight and A.codeinsight, but not xx.codeinsight or aaa.codeinsight because neither has just one character before .codeinsight. In other words, xx.codeinsight and aaa.codeinsight *will* appear in scan results if they are in your codebase.



Note • You can combine asterisks (*) and question marks (?) in your exclusion patterns.

Using Double Asterisks

Double asterisks (**) span multiple directory paths. If you create an exclusion pattern of **/codeinsight, the files in the aa/bb/cc/codeinsight directory structure will be excluded from the scan.

Sample Exclusion Patterns

The following are some sample patterns that can be used with FNCI:

Table 2-2 • Sample Exclusion Patterns and Descriptions

Pattern	Description
**/SVN/*	Excludes all the files in the SVN directories that are located anywhere in the directory tree (e.g., SVN/Repository, and apache/SVN/Entries) from the scan. But org/apache/SVN/foo/bar/Entries will be included in the scan.
/ePortal-2.0/src/**	Excludes all the files in the /ePortal-2.0/src/** directory tree (e.g., /ePortal-2.0/src/index.html, and /ePortal-2.0/src/test.xml). But /ePortal-2.0/src/**xyz.java will be included in the scan.
**/git	Exclude all files in aa/bb/cc/git.



Note • Exclusion patterns are not validated by FNCI. Please test your pattern externally.



Note • If a pattern ends with / or \, double asterisks (**) are appended. For example, codeinsight/data/ is interpreted as codeinsight/data/**.

Setting Project Defaults

The **Project Defaults** tab defines options that are global for all projects, but these can be overridden at the project level.

Currently this tab enables you to set **Task Flow Options** settings only.

Task Flow Options settings can automate the status notification, review, and remediation process for published inventory and generally work in conjunction with the policy profile associated with the project. For example, you can define the automatic creation of tasks and work items that track the review and remediation process for inventory items rejected by policy. You can also define the task flow for those items that result in a **Not Reviewed** status because policies do to apply to the items.

To override the Task Flow global options at the project level, see “Editing the Project Definition and General Settings” in the “Using FlexNet Code Insight” in the *FlexNet Code Insight User Guide*.

For more information about policies, refer to “Managing Policy Profiles” in the “Using FlexNet Code Insight” chapter in the *FlexNet Code Insight User Guide*.



Task

To set project defaults, do the following:

1. On the **FlexNet Code Insight Dashboard**, click **administration**. The **Administration** page appears with a list of side tabs.
2. Select the **Project Defaults** tab.
3. Update the **Task Flow Options** fields as needed, using the following table for field descriptions.


Table 2-3 • Task Flow fields

Column/Field	Description
When an inventory item is: impacted by a new vulnerability that violates your policy, auto-reject the inventory item	<p>This field defines what action the system should take if an inventory item is affected by a new security vulnerability (discovered during scanning or via electronic update).</p> <p>When a new security vulnerability with a CVSS score or severity <i>greater than</i> the threshold configured as policy for the Code Insight project, select this checkbox to automatically reject those project inventory items impacted by the vulnerability. (This rejection also applies to inventory items previously approved.) To indicate that an inventory item has been rejected due to new vulnerabilities, an alert icon is automatically added to the entry for each impacted inventory item on the Project Inventory tab.</p> <p>If you leave the checkbox unselected, the status of inventory items impacted by the new vulnerability remains as is.</p> <p>Note that security alerts are generated only when an electronic update, performed <i>post-scan</i>, discovers new vulnerabilities.</p> <p>For information about setting policies that define vulnerability CVSS and severity thresholds for automatic rejection or approval of inventory items, refer to “Policies Details Page” in the online help or in the <i>FlexNet Code Insight User Guide</i>.</p>

Table 2-3 • Task Flow fields (cont.)

Column/Field	Description
When an inventory item is: neither approved nor rejected by policy	<p>This field defines what action the system should take if the inventory item is <i>not</i> affected by policy (during publishing of inventory as part of a scan or manual publishing by a user).</p> <p>When Code Insight automatically publishes the inventory, define the action or action sequence that should be triggered automatically for those inventory items not automatically approved or rejected by policy:</p> <ul style="list-style-type: none"> ● take no action—Simply show the status of the inventory item as Not Reviewed on the Project Inventory tab. ● send an email notification—In addition to showing the Not Reviewed status for the inventory item, automatically send an email to the project owner, informing the project owner of the need to manually review the item. The minimum priority value affects this option. ● create a review task—In addition to showing the Not Reviewed status for the inventory item, automatically create a review task assigned to the project owner and send an email, notifying the project owner about task. (The project owner can then reassign the task to the appropriate user, such as an engineer or a legal or security expert. For details about reassigning tasks, see “Creating and Managing Tasks for Project Inventory” in the “Using FlexNet Code Insight” chapter in the <i>FlexNet Code Insight User Guide</i>.) The minimum priority value affects this option. ● create a review task with an external work item—In addition to showing the Not Reviewed status for the inventory item, perform the following: <ul style="list-style-type: none"> ● Automatically create a review task assigned to the project owner and send an email to notify the project owner about the task. (The project owner can then reassign the task to the appropriate user, such as an engineer or a legal or security expert. For details about reassigning tasks, see “Creating and Managing Tasks for Project Inventory” in the “Using FlexNet Code Insight” chapter in the <i>FlexNet Code Insight User Guide</i>.) ● Automatically associate a work item with the task, creating the work item in an Application Lifecycle Management (ALM) system (such as an issue in Jira). Code Insight creates the work item using the settings for the ALM instance to which the Code Insight project is associated. For more information about configuring an ALM instance for the project, see “ALM Settings” in the “Using FlexNet Code Insight” chapter in the <i>FlexNet Code Insight User Guide</i>. <p>The minimum priority value affects this option.</p>

Table 2-3 • Task Flow fields (cont.)

Column/Field	Description
When an inventory item is: rejected by policy	<p>This field defines what action the system should take if an inventory item is automatically rejected by policy (during publishing of inventory as part of a scan or manual publishing by a user).</p> <p>Select the action or action sequence that should be automatically triggered when an inventory item is rejected by policy:</p> <ul style="list-style-type: none"> ● take no action—Simply show the status of the inventory item as Reject on the Project Inventory tab. ● send an email notification—Automatically send an email, informing the project owner of the need to perform remediation work on the component. ● create a remediation task—Automatically create a remediation task assigned to the project owner and send an email, notifying the project owner about task. (The project owner can then reassign the task to the appropriate user, such as an engineer or a legal or security expert. For details about reassigning tasks, see “Creating and Managing Tasks for Project Inventory” in the “Using FlexNet Code Insight” chapter in the <i>FlexNet Code Insight User Guide</i>.) ● create a remediation task with an external work item—Perform the following: <ul style="list-style-type: none"> ● Automatically create a remediation task assigned to the project owner and send an email, informing the project owner about the task. (The project owner can then reassign the task to the appropriate user, such as an engineer or a legal or security expert. For details about reassigning tasks, see “Creating and Managing Tasks for Project Inventory” in the “Using FlexNet Code Insight” chapter in the <i>FlexNet Code Insight User Guide</i>.) ● Automatically associate a work item with the task, creating the work item in an Application Lifecycle Management (ALM) system (such as an issue in Jira). Code Insight creates the work item using the settings for the ALM instance to which the Code Insight project is associated. For more information about configuring an ALM instance for the project, see “ALM Settings” in the “Using FlexNet Code Insight” chapter in the <i>FlexNet Code Insight User Guide</i>.
minimum priority	<p>Select the minimum inventory priority (P1, P2, P3, or P4) to which the values for neither approved nor rejected apply.</p> <p>For example, if neither approved nor rejected by policy is set to send email notification and minimum priority is set to P3, then the email notification will only be sent out for P1, P2, and P3 inventory items that are not affected by policy. No email notification will be sent for P4 items.</p> <p></p> <p>Note • This option has no effect on the take no action value for neither approved nor rejected by policy.</p>


About FlexNet Code Insight Server REST APIs

You can create an administration client (tool) that communicates with the FlexNet Code Insight server using REST APIs to manage scan operations and to retrieve inventory information. These APIs use a REST-style interface and JSON. For more information about the Rest APIs, see the *Rest API Guide* Swagger documentation available from the **Help** menu.



Task

To view REST API documentation, do the following:

1. From any page in FlexNet Code Insight, click  and select **Help** from the menu. The **Documentation** menu appears.
2. Click **Rest API Guide**. The REST API documentation appears in a tab in your browser.
3. To view details about a particular item, click the arrow (➤) next to the item. Additional information, if available, appears under the selected item.
4. (Optional) With the details about the API visible, click the API type (GET, POST). More information about the API appears. Click **Try it out** and then click **Execute**. The application will generate cURL, make the Rest API call and display a response.

Installing & Configuring FlexNet Code Insight Plugins

FlexNet Code Insight provides plugins that you can use to perform scan activities outside of the FlexNet Code Insight user interface for integration into the Engineering and build process. This section discusses the downloading, installation, and configuration of these plugins:

- [About Plugins](#)
- [Downloading Plugins](#)
- [The Jenkins Plugin](#)
- [The JFrog Artifactory Plugin](#)
- [The Docker Images Scan Plugin](#)
- [The Bamboo Plugin](#)
- [The Maven Plugin](#)
- [The Gradle Plugin](#)
- [The Apache Ant Plugin](#)
- [The Visual Studio Team Services \(VSTS\) Extension](#)
- [The TeamCity Plugin](#)
- [The GitLab Plugin](#)

About Plugins

FlexNet Code Insight provides the following plugins that enable data (codebase files) on remote servers to be scanned:

- **Jenkins:** Allows automated scanning of a Jenkins workspace as part of the build process.
- **Artifactory:** Allows automated scanning of Artifactory repositories to identify non-compliant artifacts.
- **Docker:** Allows automated scanning of Docker images on a Docker server.
- **Bamboo:** Allows automated scanning of a Bamboo workspace as part of the build process.

- **Maven**: Allows automated scanning of Maven projects as part of the build process.
- **Gradle**: Allows automated scanning of Gradle projects as part of the build process.
- **Ant**: Allows automated scanning of Apache Ant as part of the build process.
- **Visual Studio**: Allows automated scanning of a VSTS workspace as part of the build process.
- **TeamCity**: Allows automated scanning of TeamCity projects as part of the build process.
- **GitLab**: Allows automated scanning of GitLab projects as part of the build process.
- **Generic scan plugin**: Easily integrates with other Engineering systems to perform scans as part of a build process (as described in [Scan Integration With Build Environments Through the Generic Scan Plugin](#)). It also enables you to scan arbitrary file systems of your choice or create your own scan plugin (as described in the *FlexNet Code Insight Plugin Guide*).

All the scanner plugins send results to FlexNet Code Insight for inventory creation, review, and security alerting.



Note • In addition to the scan agent plugins, a scan scheduler plugin for Jenkins is available. The scan scheduler plugin for Jenkins allows you to schedule the scan of a codebase residing on the FlexNet Code Insight server via the Jenkins scheduler.

Generating a JWT Authorization Token

FlexNet Code Insight uses a JSON Web Token (JWT) to authorize user access to the public REST API. Several of the scan agent plugins make use of the REST API, so they require a JWT. The following procedure explains how to generate and specify a JWT in FlexNet Code Insight.



Task

To generate a JWT authorization token, do the following:

1. Log into FlexNet Code Insight.
2. Open the **Admin** menu and select **Preferences**. The **Preferences** page appears.
3. Create a new JWT authorization token or copy an existing one:
 - To copy an existing JWT authorization token, click on the token name and click the **clipboard** icon (📋) that appears in the **Actions** column.
 - To create a new JWT authorization token, click **Add Token**. The **Add Token** dialog appears:
 - Type a name for the new token.
 - Pick an expiration date or select **Never Expires**.
 - Click **Save**.
4. Copy the new token string and paste it into the **Token** field on the **FlexNet Code Insight Scan** dialog to configure the desired FlexNet Code Insight scan plugin (Jenkins, Bamboo, etc.).

Downloading Plugins

The FlexNet Code Insight plugins are provided in a zip file that is not included with the product installation. You can access the plugins zip file from Flexera's Customer Community page. The following procedure assumes you have a login and password to access the Customer Community page on [Flexera.com](https://flexeracommunity.force.com/customer/).



Task

To download the plugin zip file, do the following:

1. Log into the Customer Community page of the Flexera website:
<https://flexeracommunity.force.com/customer/>
2. Click **Downloads**.
3. Click the **Access** button under **FlexNet Code Insight**. The Product and License Center page appears.
4. Select **FlexNet Code Insight** from the **Your Downloads** list.
5. Select the version of FlexNet Code Insight from the list. The **Downloads** page appears.
6. Download the CodeInsight2018Plugins.zip file.
7. When the download finishes, extract the desired plugin subdirectory to your installation directory:
 - For a standard scan plugin (such as Ant, Artifactory, Bamboo, Docker, Gradle, Jenkins, or Maven), extract the subfolder that identifies the plugin (such as code-insight-docker-images-plugin for the Docker scan plugin).
 - For the Code Insight generic scan agent plugin (required for Visual Studio Team Services, Team City, or GitLab scans), extract the subfolder code-insight-agent-sdk-generic-plugin.

Ensure that you extract the entire subfolder into your installation directory, so you have all necessary files to implement the plugin.

The Jenkins Plugin

FlexNet Code Insight provides a Jenkins build-server scan plugin to allow for automated scanning of the Jenkins workspace as part of the build process. The scan results are sent to FlexNet Code Insight for inventory creation, review, and security alerting. After the scan completes, you can log into FlexNet Code Insight, open the associated project and review any detected inventory items.

The Jenkins plugin installation and configuration process proceeds in the following manner:

1. Review the prerequisites for the Jenkins scan plugin. See [Prerequisite for the Jenkins Plugin](#).
2. Set the heap size. See [Setting Heap Size for the Jenkins Plugin](#).
3. Set up the Jenkins scan plugin. See [Setting Up the Code Insight Jenkins Plugin](#).

For examples on how to include the Code Insight scan as a part of a Jenkins Pipeline, see [Support for the Jenkins Pipeline](#).

Prerequisite for the Jenkins Plugin

Before you install and configure the Jenkins plugin, ensure that the following prerequisites are met:

- Jenkins must be installed and configured properly in your environment.
- The project must be set up in FlexNet Code Insight. For information on creating a FlexNet Code Insight project, see *Creating a Project in the FlexNet Code Insight User guide*.

Setting Heap Size for the Jenkins Plugin

The Jenkins Scan Agent plugin requires a minimum of 4GB heap for scanning. The heap size may need to be adjusted based on the number of parallel scans to be executed. In addition, ensure that you are using a 64-bit Java virtual machine (JVM) and that you run the scan agent as a Jenkins Slave, which is a Java executable that usually runs on a remote machine. The procedure for setting the heap size differs depending upon the environment you are using, Windows or Linux. Follow the procedure for your environment.

Windows



Task

To set the heap size in Windows, do the following:

1. Open the `jenkins.xml` configuration file.
2. Update the `<executable>` value to point to your 64-bit JVM:
3. Update the JVM arguments (`-Xmx` value) to allocate a minimum heap size of 4GB:

```
<executable>C:\Java\jdk1.8\jre\bin\java</executable>
```

```
<arguments>-Xrs -Xmx4g -Dhudson.lifecycle=hudson.lifecycle.WindowsServiceLifecycle -jar  
"%BASE%\jenkins.war" --httpPort=8080 --webroot="%BASE%\war"</arguments>
```



Note • The heap size may have to be adjusted based on the number of parallel scans to be executed.

Linux



Task

To set the heap size in Linux, do the following:

1. Open the `/etc/default/jenkins` file.
2. Update the JVM arguments to allocate a minimum heap size of 4GB:

```
JAVA_ARGS="-Xmx4096m"
```



Note • The heap size may have to be adjusted based on the number of parallel scans to be executed.

Setting Up the Code Insight Jenkins Plugin



Task

To set up the Jenkins plugin, do the following:

1. Extract the Jenkins scan plugin from the CodeInsight2018Plugins.zip file. For more information, see [Downloading Plugins](#).
2. Access your Jenkins server instance and navigate to **Manage Jenkins -> Manage Plugins -> Advanced tab -> Upload Plugin**.
3. Browse to the code-insight-scan-plugin.hpi file and click **Upload**.
4. Restart the Jenkins server after installing the plugin.
5. Create a new Jenkins project:
 - a. Click **New Item**.
 - b. Enter a name.
 - c. Select a project type.
 - d. Click **OK**.
6. To configure the project, select **Add post-build action** from the Post-build action dropdown menu, and select **Scan with FlexNet Code Insight**. The **Scan with FlexNet Code Insight** dialog appears.
7. Enter the following information in the **Scan with FlexNet Code Insight** dialog:
 - **Code Insight Core Server Base URL**: The URL for the core server (for example, `http://fnciserver.myorg.org:8888/codeinsight`).
 - **User Access Token**: This token generated using the FlexNet Code Insight user interface. Copy and paste the token into this field. See [Generating a JWT Authorization Token](#).
 - **Project Name**: The name of the project that was created in the FlexNet Code Insight user interface (for example, `ScanMaster_WindowsJenkins1`).
8. Click **Test Connection** to test your connection to FlexNet Code Insight.
9. Click **Save**. The next time you build, the scan will be performed after the build action.



Note • Ensure that your Jenkins server environment has a minimum of 4GB of heap space, adjusted based on the number of parallel scan to be executed. Also ensure that the environment is configured with a 64-bit JRE to support that amount of heap space. In addition, run the scan agent as a Jenkins Slave.

Support for the Jenkins Pipeline

The Code Insight plugin for Jenkins supports the inclusion of the Code Insight scan in a Jenkins Pipeline, as described in the following topics:

- [Providing the Pipeline Script for the Scan Step](#)
- [Pipeline Code Examples for Running the Scan](#)

Providing the Pipeline Script for the Scan Step

Once you build the Pipeline job, you need to include the Pipeline script for the scan step, `StartScan`, in your Pipeline code. (The next section, [Pipeline Code Examples for Running the Scan](#), provides examples of Pipeline code that include this script.)

To create the Pipeline script for the `StartScan` step, you can use one of these methods:

- Go to the Snippet Generator, select the **StartScan: Scan workspace and send results to FlexNet Code Insight** step, and generate the script. Then copy and paste the generated script into the Pipeline code.
- Simply create the script for the `StartScan` step as highlighted in the Pipeline code examples.

See [Setting Up the Code Insight Jenkins Plugin](#) for a description of the properties (base URL, project name, and JWT token) used in the Pipeline script.

Pipeline Code Examples for Running the Scan

Jenkins supports two syntax types for the development of Pipeline code:

- **Scripted syntax**—The “traditional” syntax used to develop the Pipeline as a script using Groovy as the domain-specific language.
- **Declarative syntax**—A simple, user-friendly syntax with a predefined hierarchy of statements that makes Pipeline development easier than with the Scripted syntax. Additionally, it does not require knowledge of the Groovy language. Jenkins support for the Declarative syntax was introduced with Jenkins Pipeline Plugin 2.5.

The following examples show both types of Pipeline code syntax in which the Pipeline script for the scan has been incorporated:

- [Example Declarative Pipeline Code to Run the Scan](#)
- [Example Scripted Pipeline Code to Run the Scan](#)

The Pipeline script for the scan step is highlighted in each example.

Example Declarative Pipeline Code to Run the Scan

The following is an example of Declarative code used to run the Code Insight scan as a StartScan step in the Pipeline process:

```
pipeline {
  agent any
  stages {
    stage('Checkout build and scan project1') {
      steps {
        git credentialsId: 'abcd', url: 'git://git.company.com/organization/repository1.git'
        sh "'PATH_TO_MAVEN/bin/mvn' clean install"

        StartScan (baseUrl: 'http://HOST_NAME:PORT/', projectName: 'FNCI_PROJECT_NAME',
                    token: 'JWT_TOKEN')
      }
    }
  }
}
```

Example Scripted Pipeline Code to Run the Scan

The following is an example of Scripted Pipeline code used to run the Code Insight scan as a StartScan step in the Pipeline process. The example also shows how to set up individual scans within a single Pipeline job by specifying multiple directories.

```
node {
  checkout1()
  checkout2()
}

def checkout1(){
  dir("project-1"){
    stage ('Checkout project 1'){
      git credentialsId: 'abcd', url: 'git://git.company.com/organization/repository1.git'
    }
    stage ('Build Project 1'){
      build()
    }
    stage ('Scan Project 1'){
      StartScan (baseUrl: 'http://HOST_NAME:PORT/', projectName:
                  'FNCI_PROJECT_NAME', token: 'JWT_TOKEN')
    }
  }
}

def checkout2(){
  dir("project-2"){
    stage ('Checkout project 2'){
      git credentialsId: 'abcd', url: 'git://git.company.com/organization/repository2.git'
    }
    stage ('Build Project 2'){
      build()
    }
    stage ('Scan Project 2'){

```

```

        StartScan (baseUrl: 'http://HOST_NAME:PORT/', projectName: 'FNCI_PROJECT_NAME',
                    token: 'JWT_TOKEN')
    }
}
}
def build(){
    sh "'PATH_TO_MAVEN/bin/mvn' clean install"
}

```

The Scan Scheduler Plugin for Jenkins

Before you install and configure the Jenkins Scan Scheduler plugin, ensure that the following prerequisites are met:

- Jenkins must be installed and configured properly in your environment.
- The project of interest must be set up in FlexNet Code Insight. For information on creating a FlexNet Code Insight project, see *Creating a Project in the FlexNet Code Insight User guide*.



Task

To install the FlexNet Code Insight Scan Scheduler for Jenkins, do the following:

1. Sign into Jenkins CI.
2. Navigate to **Manage Jenkins > Manage Plugins > Advanced**. The **Upload Plugin** dialog appears.
3. Click **Choose File** and select the `code-insight-scan-scheduler.hpi` file.
4. Click **Upload**.
5. Restart the Jenkins server after uploading the plugin.
6. Create a new Jenkins project:
 - Click **New Item**.
 - Enter a name.
 - Select a project type.
 - Click **OK**.
7. To configure the project, select **Add build step** from the **Build** dropdown menu, and select **Schedule a Code Insight Scan**. The **Schedule a Code Insight Scan** dialog appears.
8. Enter the following information in the **Schedule a Code Insight Scan** dialog:
 - **Server URL**: The URL for the core server. For example, `http://fnciserver.myorg.org:8888/codeinsight/`.
 - **Token**: This token must be generated in the FlexNet Code Insight Web UI and pasted into this field. See [The Bamboo Plugin](#).
 - **Project ID**: The ID of the project that was created in the FlexNet Code Insight Web UI.
9. Click **Test Connection** to test your connection to FlexNet Code Insight.
10. Click **Save**. The next time you build, the scan will be scheduled on the FlexNet Code Insight server for the configured project as part of the build.

The JFrog Artifactory Plugin

JFrog Artifactory is a binary repository manager where third-party artifacts are stored. The Artifactory repository is centralized, so all developers use the same repository to access artifacts, which provides faster access, control, and security of binary artifacts. FlexNet Code Insight provides a plugin that can scan an Artifactory repository and create inventory in a Code Insight project. Since Artifactory can contain several repositories, the plugin can also scan multiple Artifactory repositories and create inventory for several Code Insight projects.

The following topics describe how to install and use the Artifactory plugin:

- [Prerequisites for the Artifactory Plugin](#)
- [Installing the Artifactory Plugin](#)
- [Scanning an Artifactory Repository Using a Cron Job](#)
- [Scanning an Artifactory Repository Using REST API](#)
- [Scan Results](#)

Prerequisites for the Artifactory Plugin

Before installing and using the Artifactory plugin, ensure that the following prerequisites are met:

- Your site uses JFrog Artifactory PRO 5.x or higher.
- The project or projects that will contain the detected inventory items must be set up in FlexNet Code Insight. For information on creating a FlexNet Code Insight project, see “Creating a Project” in the *FlexNet Code Insight User guide*.
- The following procedures assume that you have write access for the `etc/plugins` directory on the Artifactory server. If you do not have access to that directory, be sure to obtain access before attempting to install the plugin.

Installing the Artifactory Plugin

The Artifactory plugin is available from the Downloads section of the Flexera Customer Care website. Use the following steps to install the plugin.



Task

To install the Artifactory scan plugin:

1. Extract the Artifactory scan plugin from the `CodeInsight2018Plugins.zip` file that was downloaded from the Flexera Customer Care Website. For more information about downloading plugins, see [Downloading Plugins](#).
2. Copy the following plugin directory and files into the `<artifactory_home>/etc/plugins` directory on the Artifactory server:
 - `libs` directory
 - `code-insight-scan-plugin.groovy` file
 - `code-insight-scan.plugin.props` file
3. Define the properties in the `code-insight-scan.plugin.props` file:


```
repoKeys=<repository_path1>/,<repository_path2>
```

```
codeinsight.server= http(s)://<host>:<port>/
codeinsight.auth.token=Bearer <JWT_token>
codeinsight.project.name= <project_name1>,<project_name2>
plugin.root.path=<./artifactory-pro-5.10.2/etc/plugins>
plugin.project.description= will be set by plugin, can be left blank
isScanCronJobEnabled=disabled
isPluginEnabled=enabled
cronJobTime=1 * * * * ?
artifactory_url= http(s)://<host>:<port>/artifactory/
```

4. Determine if you want to execute the scan with a cron job or by calling REST API:
 - To execute a scan with a cron job, see [Scanning an Artifactory Repository Using a Cron Job](#).
 - To execute a scan by calling REST API, see [Scanning an Artifactory Repository Using REST API](#).

Scanning an Artifactory Repository Using a Cron Job

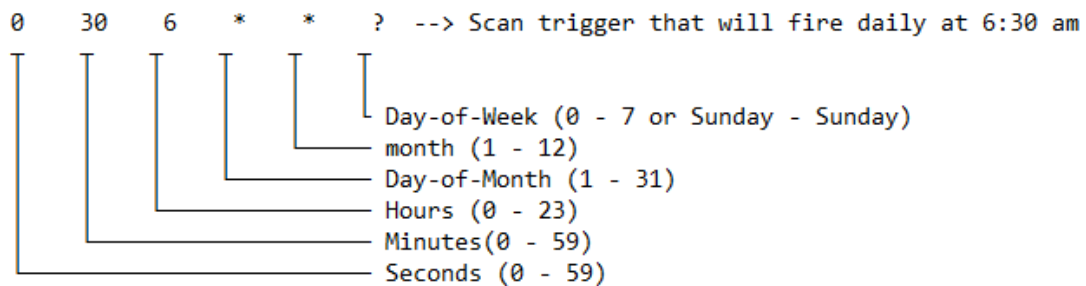
You can use the following procedure to schedule an Artifactory repository scan to run periodically.



Task

To execute an Artifactory scan using a cron job:

1. Open the code-insight-scan.plugin.props file.
2. Modify the property isScanCronJobEnabled=disabled to isScanCronJobEnabled=**enabled**.
3. Set the cronJobTime property to schedule the scan. Use the following diagram and the example it provides to help you set the property.



4. Restart the Artifactory server.

Scanning an Artifactory Repository Using REST API

You can call REST API to scan all Artifactory repositories listed in the code-insight-scan.plugin.props file or to scan a specific repository instead. The following topics describe how to scan repositories using REST API:

- [Requirements When Using REST API to Scan Repositories](#)
- [Scanning All Repositories](#)
- [Scanning a Specific Repository](#)
- [Reloading the Artifactory Plugin](#)

Requirements When Using REST API to Scan Repositories

The following lists the requirements for using REST APIs to scan Artifactory repositories.

Prerequisite for Scanning Repositories

As prerequisite for using REST API to scan Artifactory repositories, ensure that the properties in `code-insight-scan.plugin.props` are properly defined according to the instructions in [Installing the Artifactory Plugin](#) and according to any specific instructions listed in the procedures.

Required Option When Using the “https” Protocol

The REST API calls used in the next sections use the `http` protocol. To use the `https` protocol instead, be sure to include the option `-k` in the call:

```
curl -X POST -u<user_name>:<password> -k "https://<artifactory_host>:8081/artifactory/api/plugins/execute/CodeInsightScan"
```

Scanning All Repositories

The following command scans all repositories listed in the `code-insight-scan.plugin.props` file.



Task

To scan all repositories

Use the following API call to scan all repositories:

```
curl -X POST -u<user_name>:<password> "http://<artifactory_host>:8081/artifactory/api/plugins/execute/CodeInsightScan"
```

Scanning a Specific Repository

The following procedure scans a specific repository.



Task

To scan a specific repository:

1. Ensure that the following properties are also defined in the `code-insight-scan.plugin.props` file: `codeinsight.server`, `codeinsight.auth.token`, `plugin.root.path`, `isPluginEnabled`, and `artifactory_url`.
2. Use the following API call to scan the repository:

```
curl -X POST -u<user_name>:<password> "http://<artifactory_host>:8081/artifactory/api/plugins/execute/CodeInsightSingleScan?params=repoKey=<repository_name>%7cproject=<FNCCI_project_name>"
```

Reloading the Artifactory Plugin

If you have downloaded an updated version of the Code Insight plugin for Artifactory, you can use this REST API call to reload the plugin before running a scan:

```
curl -X POST --u<user_name>:<password> http://localhost:8081/artifactory/api/plugins/reload
```

Scan Results

When the scan completes, inventory is created in the corresponding FlexNet Code Insight project. The **Scan Status** section on the **Project Summary** page provides information about the scan.

Similarly in Artifactory, information about the scan, such as the Code Insight project name, the scan status, and a link to the Code Insight project inventory are provided for each repository scanned. For more information about using plugins in Artifactory, see the following site:

<https://www.jfrog.com/confluence/display/RTF/User+Plugins>

The Docker Images Scan Plugin

Docker is a containerization tool that packages applications and their dependencies into containers, which are comprised of static images. These images are themselves comprised of layers. FlexNet Code Insight provides a plugin to allow the scanning of Docker images on a Docker server.



Note • It is recommended that Docker images be scanned on a development, test, or staging server before being pushed to a production instance as part of the DevOps process flow.

Installing and Launching the Docker Images Plugin

Before you install and configure the Docker Images plugin, ensure that the following prerequisites are met:

- The Docker server must be installed and configured properly in your environment. The Docker scan plugin can only be executed on a server that already has an authenticated connection to the Docker server.



Note • The Docker scan plugin issues Docker commands without prompting for credentials.

- The project that will contain the detected inventory items must be set up in FlexNet Code Insight. For information on creating a FlexNet Code Insight project, see “Creating a Project” in the *FlexNet Code Insight User guide*.
- A minimum of 2GB of heap space must be allocated on the Docker server, which must be configured with a 64-bit JRE to support that amount of heap space.



Task

To install and launch the Docker scan plugin, do the following:

1. Extract the Docker images scan plugin from the CodeInsight2018Plugins.zip file. For more information, see [Downloading Plugins](#).

2. Open the code-insight.docker.props file in a text editor:

```
//required
codeinsight.server=http://127.0.0.1:8888
codeinsight.auth.token=Bearer
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbiIsInVzZXJJZCI6MSwiaWF0IjoxNTExNDM1MTk4fQ.dHItJjJ2c89Dg5cVLvf
GR3fwJcR3yA1VE6k98dRZTdp3h6McDgv_PloVVE88eJ2GOG0tND0nhU0ShDLUzdu3Pg
codeinsight.project.name=inv2
plugin.root.path=/Users/ranimathur/Work/Scratch/
//optional
plugin.project.name=plugin project name
plugin.project.description=plugin project description
plugin.path.prefix=$demo_workspace/
```

3. Edit the code-insight.docker.props file to specify the following information:

- codeinsight.server (required): The URL path to the FlexNet Code Insight server.
- codeinsight.auth.token (required): The JWT authentication token that you obtain from the FlexNet Code Insight server using the Bearer schema. See [Generating a JWT Authorization Token](#).
- codeinsight.project.name (required): The name of the FlexNet Code Insight project.
- plugin.root.path (required): The root path where the docker plugin will be executing. This path must have writable privileges for the user executing the plugin.
- plugin.project.name (optional): A more descriptive name to the project being scanned, that may be different from the project name specified in the FlexNet Code Insight server. This text will appear in the **Project Summary** page of the FlexNet Code Insight GUI.
- plugin.project.description (optional): A description of the project being scanned. This text will appear in the **Project Summary** page of the FlexNet Code Insight GUI.
- plugin.path.prefix (optional): The path prefix of the image being scanned. This prefix will be used to reference the file paths of the codebase on the **Project Inventory** page of the FlexNet Code Insight GUI.

4. Issue the following command to launch the Docker plugin from the command line:

```
% code-insight-docker-plugin.sh -image <docker_image_name>
```

The docker_image_name is the name given to the image that FlexNet Code Insight is to scan.



Note • Only the downloaded Docker image is scanned.

As it runs, the Docker scan plugin does the following:

- Contacts the FlexNet Code Insight server to validate the connection and download a scanner.
- Issues the Docker commands to download the Docker image.
- Extracts the Docker image.

- Scans the extracted Docker image contents.

The plugin sends the inventory results to FlexNet Code Insight configured.



Note • The Docker scan plugin must be launched whenever the Docker image is updated. The Docker scan plugin can be included in a script, so the image is scanned regularly.

The Bamboo Plugin

FlexNet Code Insight provides a Bamboo build-server scan plugin to allow automated scanning of a Bamboo workspace as part of your application build process. The scan results are sent to FlexNet Code Insight for inventory creation, review, and security alerting. The Bamboo plugin scans only the application root folder.

Before you install and configure the FlexNet Code Insight Bamboo plugin, ensure that the following prerequisites are met:

- Bamboo 5.2 or higher must be installed and configured as explained in the Bamboo Installation guide.
- Minimum heap memory size must be set to 4GB for the Bamboo Server.
- An inventory-only project must be set up in FlexNet Code Insight. For information, see *Creating a Project* and *Performing Inventory-only Scanning* in the FlexNet Code Insight User guide or online help.

Installing & Configuring the Bamboo Plugin

The following procedure covers installing and configuring the Bamboo plugin, which requires you to perform actions in both Bamboo and FlexNet Code Insight.



Task

To install and configure the Bamboo plugin, do the following:

1. Extract the Bamboo scan plugin from the CodeInsight2018Plugins.zip file. For more information, see [Downloading Plugins](#).
2. Access your Bamboo server instance.
3. From the **Bamboo Administration icon**, click **Add-ons**.
4. Click **Upload add-on**.
5. Browse to the code-insight-bamboo-scan.jar and click **Upload**. The Bamboo jar file is located wherever the zip file containing the plugins was extracted.
6. Create a project in Bamboo:
 - Create the plan.
 - Add a job.
 - Add a FlexNet CodeInsight Scan task to a build project.
7. If you have not done so, create an inventory-only project with a name corresponding to the Bamboo project. For information about creating an inventory-only project, see *Performing Inventory-only Scanning* in the FlexNet Code Insight User guide or online help.

8. Enter the following information in the FlexNet CodeInsight Scan task:
 - **Server URL:** The URL for the core server. For example, <http://fnciserver.myorg.org:8888/codeinsight/>.
 - **Token:** This token must be generated in the FlexNet Code Insight Web UI and pasted into this field. See [Generating a JWT Authorization Token](#).
 - **Project ID:** The ID of the project that was created in the FlexNet Code Insight Web UI.
9. Click **Save**. If the Server URL and Token value are correct, the task will be saved. The next time you run the plan, the automated scan of the workspace will be executed for the configured project as part of the plan.



Note • The scan task should be placed after the build task in the plan's task sequence.

The Maven Plugin

Maven is a tool that simplifies the building and management of Java-based projects. The FlexNet Code Insight Maven scan plugin allows you to scan an application project during its build on Maven without disrupting the established build process. Once scanned, the codebase can be analyzed in the FlexNet Code Insight user interface. The Maven scan plugin makes it easy to incorporate scanning and analysis into your development workflow.

For more information, refer to the following:

- [More About the Maven Scan Plugin](#)
- [Prerequisites for the Maven Scan Plugin](#)
- [Installing and Configuring the Maven Scan Plugin](#)
- [Cleaning the Application Project](#)
- [Running the Maven Goal for the Code Insight Scan](#)

More About the Maven Scan Plugin

The Code Insight Maven scan plugin scans only the following items:

- Direct dependencies of a project
- Transitive dependencies of a project
- Build folder containing the application jars

The plugin creates a Maven goal called `code-insight-scan`, which will be executed along with the *install* phase of the build cycle to get inventory details, as described later in [Running the Maven Goal for the Code Insight Scan](#).

Prerequisites for the Maven Scan Plugin

Before you install and configure the Code Insight Maven scan plugin, ensure that the following prerequisites are met:

- JDK 1.8 is installed.
- Maven is installed.

- %MAVEN_HOME%/bin is configured and added to the path environment variable. (This prerequisite avoids SSL certification issues.)
- A JWT token has been generated to authorize calls to FlexNet Code Insight. See [Generating a JWT Authorization Token](#).

Installing and Configuring the Maven Scan Plugin

Use the following steps to install and configure the Code Insight Maven scan plugin.



Task

To install and configure the Code Insight Maven scan plugin:

1. From the CodeInsight2018Plugins.zip file that was downloaded from the Flexera Customer Care Website, extract the Maven scan plugin subdirectory (code-insight-maven-plugin) to a location on your local disk. The recommended location to which to extract this subdirectory is the application project directory.
2. Execute the following commands to install the plugin into the Maven local repository:

```
mvn install:install-file -Dfile="$<PROJECT_DIRECTORY>/code-insight-maven-plugin/lib/code-insight-maven-scan-<VERSION>.jar" -DpomFile="$<PROJECT_DIRECTORY>/code-insight-maven-plugin/lib/pom.xml" -DgroupId=com.flexnet.maven -DartifactId=code-insight-maven-scan -Dversion=<VERSION> -Dpackaging=jar
```

```
mvn install:install-file -Dfile="$<PROJECT_DIRECTORY>/code-insight-maven-plugin/lib/codeinsight-agent-<VERSION>.jar" -DgroupId=com.flexnet.codeinsight -DartifactId=codeinsight-agent -Dversion=<VERSION> -Dpackaging=jar
```

Note the following variables:

- \$<PROJECT_DIRECTORY> is your application project directory (or the local directory to which you extracted the plugin).
 - <VERSION> is the latest version of specific jar file referenced in the command (either the code-insight-maven-scan or codeinsight-agent jar file).
3. Add the following information to your application pom.xml file. Refer to [Plugin and FNCI Server Settings](#) for a description of the values you need to provide for the plugin and fnciServerSettings sections.

```
<plugin>
  <groupId>com.flexnet.maven</groupId>
  <artifactId>code-insight-maven-scan</artifactId>
  <version>latest_codeinsight_maven_scan_jar_version</version>
  <inherited>>false</inherited>
  <executions>
    <execution>
      <phase>install</phase>
      <goals>
        <goal>code-insight-scan</goal>
      </goals>
    </execution>
  </executions>
  <configuration>
    <fnciServerSettings>
      <fnciServer>server_url</fnciServer>
      <fnciAuthToken>bearer_server_token_value</fnciAuthToken>
    </fnciServerSettings>
  </configuration>
</plugin>
```

```

    <fnciProjectName>FNCI_project_name</fnciProjectName>
    <pluginRootPath>plugin_root_path</pluginRootPath>
    <pluginProjectName>plugin_project_name</pluginProjectName>
    <pluginDescription>any_plugin_description</pluginDescription>
    <pluginPathPrefix>plugin_path_prefix</pluginPathPrefix>
  </fnciServerSettings>
</configuration>

</plugin>

```

Plugin and FNCI Server Settings

The following describes the settings that you need to define in the `plugin` and `fnciServerSettings` sections of the information you are adding to the application `pom.xml` file (as described in Step 3 of the previous procedure).

Table 3-1 • FNCI Server Settings in the Application “pom.xml” File

Setting	Description
version	The version of the <code>code-insight-maven-scan-<version>.jar</code> file included with the current scan plugin (for example, 1.0.2).
fnciServer	(Required) The URL path to the FlexNet Code Insight server in the following format: <code>http://<Code_Insight_server_host_name>:<port_number>/codeinsight/</code>
fnciAuthToken	(Required) The JSON Web Token (JWT) used to authorize user access to the Code Insight server functionality. The token format includes the command Bearer , followed by the token value, as in the example: Bearer eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbiIsInVzZXJJZCI6MSwia For information about generating the JWT, see Generating a JWT Authorization Token .
fnciProjectName	(Required) The name of the FlexNet Code Insight inventory-only project created on the Code Insight server for your application codebase scans.
pluginRootPath	Currently not used.
pluginProjectName	(Optional) The name of the <i>application project</i> being scanned. This name will appear, along with the Code Insight project name, in the Last Scan field on the Project Summary page in the FlexNet Code Insight user interface. It provides a reference to help a reviewer or developer identify what codebase was scanned.
pluginDescription	(Optional) A description of the application project being scanned. This text will appear in the Description field on the Project Summary page in the FlexNet Code Insight user interface.

Table 3-1 • FNCI Server Settings in the Application “pom.xml” File

Setting	Description
pluginPathPrefix	(Optional) The path prefix for the codebase files being scanned. This prefix is used to reference the codebase file paths on the Project report generated from the Project Summary page in the FlexNet Code Insight user interface.

Cleaning the Application Project

During a build, Maven can cache a great deal of output. This cached output can have a negative impact on the performance of the Code Insight Maven scan plugin. Therefore, before you run the Maven goal for the Code Insight scan, it is recommended that you clean the application project, a process that clears the cache of the artifacts of previous builds.



Task *To clean the application project:*

Execute the following command:

```
mvn clean
```

Running the Maven Goal for the Code Insight Scan

After you clean the application project, you can run the `code-insight-scan` Maven goal, which will scan the codebase.



Task *To execute the goal that runs the Code Insight scan:*

To build the application (and run the Code Insight scan as part of the build cycle), execute the following command:

```
mvn install
```

Alternatively, to execute the Code Insight scan only, run the specific goal:

```
mvn code-insight-maven-scan:code-insight-scan
```

The Gradle Plugin

Gradle is a build automation system that uses the Groovy language to establish the configuration of the build project, rather than using XML as Maven does. The FlexNet Code Insight Gradle plugin allows a codebase created in Gradle to be scanned and then analyzed in the FlexNet Code Insight user interface.

The Gradle plugin scans only the following items:

- Direct dependencies of a project.
- Transitive dependencies of a project.
- The distribution folder containing application jars.



Note • The Gradle plugin does not scan the jars present in the `lib` folder, which are plugin-dependent jars.

Before you install and configure the FlexNet Code Insight Gradle plugin, ensure that the following items are correctly installed and configured:

- JDK1.8
- Gradle
- JWT token to authorize calls to FlexNet Code Insight. See [Generating a JWT Authorization Token](#).

Installing and Configuring the Gradle Plugin

To use the Gradle plugin, you must add the settings to the application's `build.gradle`. This section contains the procedure for installing and configuring the Gradle plugin.



Task

To install the Gradle plugin, do the following:

1. Extract the Gradle plugin from the `CodeInsight2018Plugins.zip` file. See [Downloading Plugins](#).
2. Add all the dependent jars in the `code-insight-gradle-plugin` to the application class path by doing the following:
 - Create a folder named `dependent_jars` within the application.
 - Copy all jar files into that folder.
 - Add the following configuration to make the jars available to the classpath:

```
buildscript {
    dependencies {
        classpath files(fileTree(dir: 'dependent_jars', includes: ['*.jar']))
    }
}
```

3. Add the following settings to the `build.gradle` file:

```
apply plugin: 'code-insight-gradle-plugin'

scanSettings {
    fnciServer= "server_url"
    fnciAuthToken= "Bearer server_token_value"
    fnciProjectName= "provide FNCI project name here"
    pluginRootPath= "provide/plugin/root/path/here"
    pluginProjectName= "provide plugin project name here"
    pluginDescription= "provide any plugin description here"
    pluginPathPrefix= "provide/plugin/path/prefix"
}
```

Where:

`scanSettings` is an extension to provide the FlexNet Code Insight scan server settings.

`fnciServer` is the hosted server where the FlexNet Code Insight application is running. This field is required.

fnciAuthToken is the user identification token of the FlexNet Code Insight server containing the command “Bearer”, followed by the token value. For example, *Bearer eyJhbGciOiJIUzUxMiJ9.eyJzdWl0OiJhZG1pbilzInVzZXJJZC6MSwia*. This field is required. For information about generating the Auth token, see [Generating a JWT Authorization Token](#).

fnciProjectName is the project name that you are running the plugin against. This project name (inventory-only type) must be present on the FlexNet Code Insight server. For example, *test 1*. This field is required.

pluginRootPath is the path where the plugin will be launched (usually the root of the application). For example, *D:\\test\\Gradle_test\\Gradle_application*. This field is required.

pluginProjectName is where you can provide the name of the application against which you are running the plugin. For example, *Gradle Application*. This field is optional.

pluginDescription is a description of the application, which will be seen on the Project Summary page in FlexNet Code Insight. For example, *Gradle Application Test*. This field is optional.

pluginPathPrefix inventories the file path prefix value on the FlexNet Code Insight server, which can be seen in the **Associated Files** section of inventory in FlexNet Code Insight. For example, *demo_workspace/*. This field is optional.

4. To run a scan of the application, use the `code-insight-scan` task as in one of the following:

- After the build, run `gradle code-insight-scan`.

or

- With the build, run `gradle build code-insight-scan`.

The Apache Ant Plugin

Apache Ant is a tool to support the build process for Java projects. Ant is often used in conjunction with other build tools such as Maven. FlexNet Code Insight provides a plugin that is executed along with the target of the build cycle to obtain inventory details. The Apache Ant plugin scans only the application root folder.

Before you install and configure the FlexNet Code Insight Ant plugin, ensure that the following items are correctly installed and configured:

- JDK 1.8 installed.
- Apache Ant installed.
- To authorize calls to FlexNet Code Insight, see [Generating a JWT Authorization Token](#).

Configuring the Plugin



Task

To configure the plugin, do the following:

1. Extract the Ant plugin from the `CodeInsight2018Plugins.zip` file. See [Downloading Plugins](#).
2. Configure `%ANT_HOME%` and add `%ANT_HOME%/bin` to the path variable.folder.
3. To check the Ant plugin installation, run the following command:

```
ant -v
```

4. Add all the dependent jars from the code-insight-ant-plugin folder to the application's compile classpath.
5. To run the task `codeinsightantplugin` along with the compile target, paste the taskdef code snippet into the compile target and run the following command:

```
ant compile
```

For the code snippet, see [Executing the Scan](#).

6. Copy the `code-insight-ant.jar` into the path used for the compile task, and set the `classpathref` of the `javac` task as the `classpathref` in the `codeinsightantplugin` task.

Executing the Scan



Task

To execute the scan, do the following:

1. Run the following command:

```
ant targetname
```

For example: `ant compile`

2. To execute the scan along with any target of the build lifecycle, apply the plugin inside the target in the `build.xml` of the Ant application as follows:

```
<taskdef name="codeinsightantplugin" classname="com.ant.plugin.CodeInsightAntPlugIn"
classpath=" " classpathref=" " />
<codeinsightantplugin fnciServer="server_url" fnciauthtoken="Bearer server_token_value "
fnciprojectname="provide FNCI project name here"
pluginRootPath="provide/plugin/root/path/here"
pluginProjectName="provide plugin project name here"
pluginindescription="provide any plugin description here"
pluginPathPrefix="provide/plugin/path/prefix">
</codeinsightantplugin>
```



Note • Although specifying `taskdef.classpath` is not mandatory, you should set the `path id` of the `javac` task as the `Classpathref` in the `codeinsightantplugin` taskdef. If the application does not have a `javac path-id` defined in the `build.xml`, you must define one new `path id` referring to all compile time dependencies and use this as `Classpathref`. For example:

```
<path id="cp" <fileset dir="lib">
<include name="*.jar" />
</fileset>
</path>
```

Use `"cp"` as the `Classpathref` in the taskdef.



Note • The Ant plugin project name can not include the ampersand (&) character.

The Visual Studio Team Services (VSTS) Extension

The Visual Studio Team Services (VSTS) extension for FlexNet Code Insight allows development teams to easily integrate open source scanning into their build. The scan results are sent to FlexNet Code Insight for inventory creation, review, and security alerting.

To enable this functionality, you need to install the FlexNet Code Insight Scan extension and configure the build process to include the scan:

- [Prerequisite](#)
- [Installing the FlexNet Code Insight VSTS Extension](#)
- [Adding a FlexNet Code Insight Scan Task to Your Agent Job](#)

Prerequisite

Before you install and configure the FlexNet Code Insight Scan extension, create a project in Code Insight with a **Project Type of Inventory-Only**. (This project type is used because the plugin allows for remote scanning and does not require that you upload your codebase to Code Insight.) After the scan is finished, you can log into Code Insight and review the detected inventory items in the project.

For instructions on creating a Code Insight project, refer to “Creating a Project” in the *FlexNet Code Insight User Guide*.

Installing the FlexNet Code Insight VSTS Extension

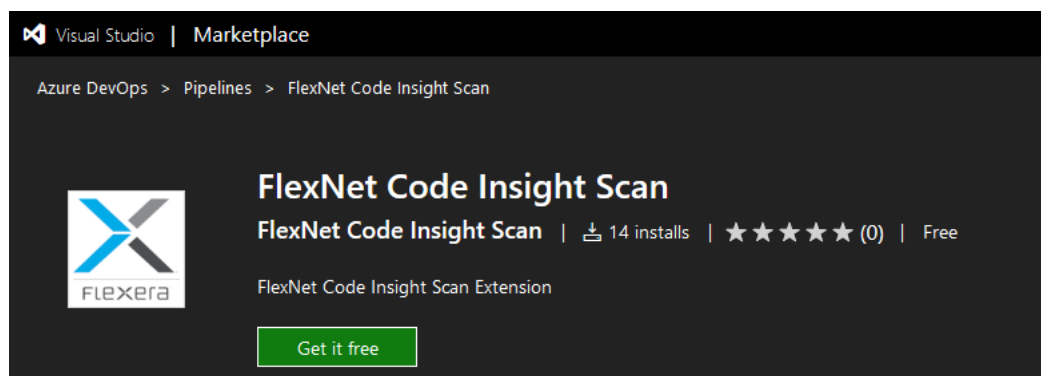
To obtain and install the FlexNet Code Insight Scan extension, perform the following steps.



Task

To obtain and install the FlexNet Code Insight Scan extension:

1. Open the Visual Studio Marketplace:
<https://marketplace.visualstudio.com/>
2. In the **Visual Studio Team Services** section, search for the **FlexNet Code Insight Scan** extension.



3. Download and install this extension into Visual Studio.

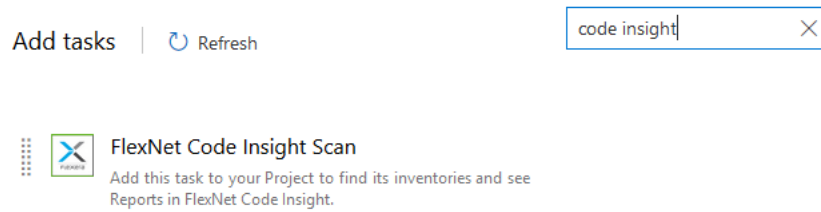
Adding a FlexNet Code Insight Scan Task to Your Agent Job

After the **FlexNet Code Insight Scan** extension has been installed, you need to add a FlexNet Code Insight Scan task to your agent job so that the scan is automatically performed as part of your build process.

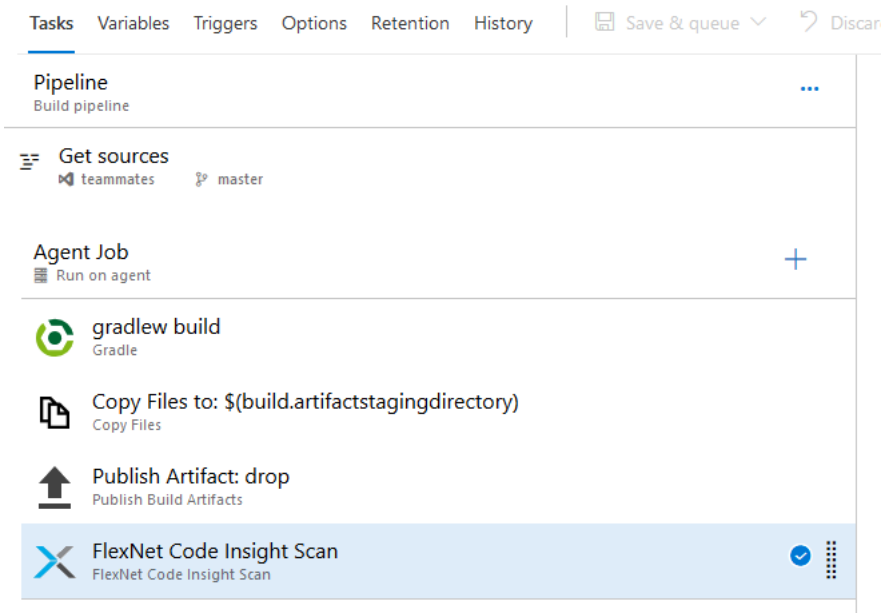


Task To add a FlexNet Code Insight Scan task to your Agent job:

1. Create a build pipeline for your Visual Studio project.
2. Locate the **FlexNet Code Insight Scan** task in the task catalog.



3. Add the **FlexNet Code Insight** task at any point after the build task has completed.



4. Define the scan task properties on the **FlexNet Code Insight Scan** window:

FlexNet Code Insight Scan ⓘ

Link settings View YAML Remove

Version 1.*

Display name *
FlexNet Code Insight

FlexNet Code Insight Server * ⓘ
http://MY_SERVER:8888

Authentication Token * ⓘ
MY_TOKEN

FlexNet Code Insight Project Name * ⓘ
MY_PROJECT

Folder(s) to Scan ⓘ
\$(build.artifactstagingdirectory)

Control Options

Output Variables

The following describes the task properties:

Field	Description
FlexNet Code Insight Server	The URL for the core server (for example, http://fnciserver.myorg.org:8888/codeinsight/). Ensure that the URL is publicly accessible and that the port is available.
Authorization Token	The JWT authorization token generated using the FlexNet Code Insight user interface. (Copy and paste the token into this field.) See Generating a JWT Authorization Token .
FlexNet Code Insight Project Name	The name of the inventory-only project that was created in the FlexNet Code Insight user interface (for example, ScanProject2_VSTS).
Folder(s) to Scan	The folder containing the code to scan. Typically, you would use \$(build.artifactstagingdirectory) , which is the location where the build output is staged during the build process.

5. Save and queue the build definition.

The scan will be performed in the build environment as part of the build process, and the results will be sent to the configured FlexNet Code Insight project. The resulting inventory items can be viewed in the FlexNet Code Insight user interface.

Scan Integration With Build Environments Through the Generic Scan Plugin

FlexNet Code Insight includes a generic scan plugin that allows integration of FlexNet Code Insight with various Engineering applications for automatic composition scanning as part of the build process. The Visual Studio Team Services (VSTS) Private Agent Server, TeamCity, GitLab integrations utilize the generic scan plugin for remote scanning and returning the results back to Code Insight for further review and reporting.

The discovered inventory items are created on the Code Insight server can be reviewed automatically via policies or manually reviewed by various stakeholders. Security alerts with corresponding email notifications will be generated for any inventory items with new security vulnerabilities.

See the next section, [Downloading the Generic Scan Plugin](#), for information about downloading the Code Insight generic scan plugin and about its prerequisites. For more about the Code Insight generic plugin, including how to use it to scan arbitrary file systems of your choice or to create a custom scan plugin, see the *FlexNet Code Insight Plugin Guide*.

Refer to the following topics about how to obtain the Code Insight generic scan plugin and prepare to use it:

- [Downloading the Generic Scan Plugin](#)
- [Prerequisites for Using the Generic Scan Plugin](#)

Downloading the Generic Scan Plugin

Refer to [Downloading Plugins](#).

Copy (or directly extract) the `code-insight-agent-sdk-generic-plugin` folder to the location specified in the appropriate section for the given build system, as described below.

Prerequisites for Using the Generic Scan Plugin

The following prerequisites are required to use the FlexNet Code Insight generic scan plugin:

- A minimum of 4 GB heap is required for scanning.
- A FlexNet Code Insight **Inventory Only** project needs to be created to store the discovered inventory items. (Refer to the “Creating a Project” section in the *FlexNet Code Insight User Guide*.)
- Internet access is not required, but is recommended.
 - If Internet access is available, the scan agent will periodically download the latest security vulnerability definitions from the National Vulnerability Database (NVD).
 - If Internet access is not available, then the default signatures that were released with the latest version of FlexNet Code Insight will be used.

Any additional requirements for a given build environment are addressed in the appropriate section for that build environment (VSTS, TeamCity or GitLab), as follows.

The TeamCity Plugin

This section explains how to configure TeamCity to integrate with the FlexNet Code Insight generic scan plugin to automatically perform composition scanning as part of the build process. The scan occurs on the TeamCity build agent. The following topics are covered:

- [Prerequisites](#)
- [Installing the Generic Scan Agent on TeamCity Agent Configured on Windows](#)
- [Configuring a Build to Run a Code Insight Scan](#)
- [Executing the Build](#)

Prerequisites

The following prerequisites are required to integrate TeamCity with the FlexNet Code Insight generic scan plugin:

- All the prerequisites listed in [Prerequisites for Using the Generic Scan Plugin](#).
- A TeamCity build agent needs to be installed and configured to use the FlexNet Code Insight generic scan plugin. (Refer to <https://confluence.jetbrains.com/display/TCD10/Setting+up+and+Running+Additional+Build+Agents> for instructions.)

Installing the Generic Scan Agent on TeamCity Agent Configured on Windows

The FlexNet Code Insight generic scan plugin is located in the `code-insight-agent-sdk-generic-plugin` folder that you extracted from the `CodeInsight2018Plugins.zip` file. Copy (or directly extract) this folder to the TeamCity build agent (see [Downloading Plugins](#)).

The scan plugin folder contains a jar file and a sample batch and shell script.

On the Team City build agent, update the following to match your environment:

- The codebase root:

```
SET ROOT_PATH=C:\Codebase\output
```
- The bin folder location for the generic scan plugin:

```
cd C:\agent\GenericScanPlugin\example\bin
```

Note that the first time a scan is performed using the generic scan plugin, a data snapshot is downloaded from the National Vulnerability Database (NVD) to generate an index of the latest security vulnerabilities.

Configuring a Build to Run a Code Insight Scan

Follow these steps to configure a build to run a Code Insight scan.



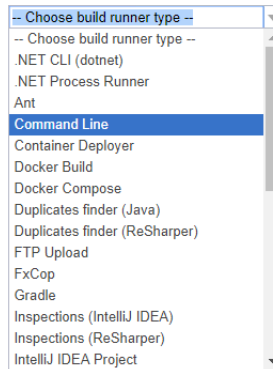
Task

To configure a build to run a Code Insight scan:

1. Log into TeamCity, select your project, and create a new Build Configuration.
2. To configure a build step to run a Code Insight scan, select on your build configuration, and click **Add Build Step**.
3. From the **Runner type** list, select **Command Line**.

New Build Step: | ▾

Runner type:



4. Configure the **Command line** build step for the Code Insight scan:

- a. Enter a value for **Step name** (for example, **Codeinsight Scan**) to identify the step.
- b. In the **Run** field, select **Custom script**.
- c. In the **Custom script** field, provide the following:

```
C:\GenericScanPlugin\example\bin\TeamCity_FNCIScan.bat < FNCI_PROJECT> <FNCI_SERVER> <AUTH_TOKEN> <SCAN_DIR>
```

Replace the following variables in the script with the appropriate information:

- <FNCI_PROJECT> with the name of the project you created earlier to capture the inventory
- <FNCI_SERVER> with your Code Insight server URL (for example, <http://1.1.1.1:8888/codeinsight>)
- <AUTH_TOKEN> with your JWT authorization token obtained from the Code Insight server (as described in [Generating a JWT Authorization Token](#))

 A screenshot of the 'Build Step' configuration dialog in TeamCity. The 'Runner type' is set to 'Command Line'. The 'Step name' is 'Code Insight Scan'. The 'Run' dropdown is set to 'Custom script'. The 'Custom script' field contains the command: 'C:\GenericScanPlugin\example\bin\TeamCity_FNCIScan.'. Below the field, there is a note: 'A platform-specific script, which will be executed as a .cmd file on Windows or as a shell script in Unix-like environments.' At the bottom, there are 'Save' and 'Cancel' buttons.

When complete, your build configuration should look like this:

Build Steps
In this section you can configure the sequence of build steps to be executed. Each build step is represented by a build runner and provides integration with a specific build or test tool.

[+ Add build step](#)

Build Step	Parameters Description		
1. Code Insight Scan	Command Line Custom script: C:\GenericScanPlugin\example\bin\TeamCit... Execute: If all previous steps finished successfully	Edit	

Executing the Build

The next time your build is executed, a FlexNet Code Insight agent scan will be performed at the end of the build process. If you have scheduled the Code Insight scan job, after a Maven build, for example, you should see something like below in your TeamCity build queue:

▼ Maven Code Insight Scan ▼	no hidden ▼	×
▼ Build ▼		Run ... ▼
#2	Tests passed: 836 ▼	No artifacts ▼ 38851806+vdonga (1) ▼ 9 minutes ago (1m:52s)
▼ Code Insight Scan ▼		Run ... ▼
#1	Success ▼	No artifacts ▼ No changes ▼ 5 minutes ago (4m:36s)

The GitLab Plugin

This section explains how to configure GitLab to integrate with the FlexNet Code Insight generic scan plugin to automatically perform composition scanning as part of the build process. The scan occurs on the GitLab runner. The following topics are covered:

- [Prerequisites](#)
- [Installing the Generic Scan Agent on GitLab Runner Configured on Windows](#)
- [Configuring the CI/CD Pipeline to Run a Code Insight Scan](#)
- [Executing the Build](#)

Prerequisites

The following prerequisites are required to integrate GitLab with the FlexNet Code Insight generic scan plugin:

- All the prerequisites listed in [Prerequisites for Using the Generic Scan Plugin](#).
- A GitLab runner needs to be installed and configured to use the FlexNet Code Insight generic scan plugin. (Refer to <https://docs.gitlab.com/runner/install/> for instructions.)

Installing the Generic Scan Agent on GitLab Runner Configured on Windows

The FlexNet Code Insight generic scan plugin is located in the code-insight-agent-sdk-generic-plugin folder that you extracted from the CodeInsight2018Plugins.zip file. Copy (or directly extract) this folder to the GitLab runner (see [Downloading Plugins](#)).

The scan plugin folder contains a jar file and a sample batch and shell script.

On the GitLab runner, update the following to match your environment:

- The codebase root:
`SET ROOT_PATH=C:\GitLab-Runner\output`
- The bin folder location for the generic scan plugin:
`cd C:\GitLab-Runner\GenericScanPlugin\example\bin`

Note that the first time a scan is performed using the generic scan plugin, a data snapshot is downloaded from the National Vulnerability Database (NVD) to generate an index of the latest security vulnerabilities.

Configuring the CI/CD Pipeline to Run a Code Insight Scan

To configure the CI/CD pipeline in your GitLab project to scan for FlexNet Code Insight scan, you need to edit your `.gitlab-ci.yml` file.



Task

To edit the `.gitlab-ci.yml` file:

Add the following contents to the file:

```
variables:
  FNCI_SERVER: <FNCI_SERVER>
  FNCI_TOKEN: <AUTH_TOKEN>
  FNCI_PROJECT: <FNCI_PROJECT_NAME>

codeinsight_scan:
  stage: test
  only:
    - master
  tags:
    - <tag for your GitLab-Runner>
  script:
    - cmd /Q /C C:\Gitlab-runner\GenericScanPlugin\example\bin\run_scan.bat %FNCI_PROJECT%
      %FNCI_SERVER% %FNCI_TOKEN% %CI_PROJECT_DIR%
```

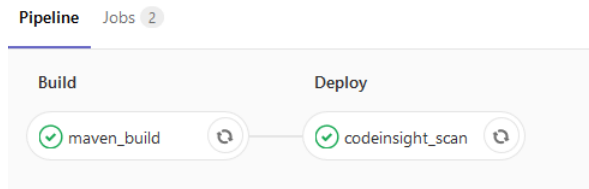
Replace the following variables with the appropriate information:

- <FNCI_SERVER> with your Code Insight server (for example, <http://1.1.1.1:8888/codeinsight>)
- <AUTH_TOKEN> with your JWT authorization token obtained from the Code Insight server (as described in [Generating a JWT Authorization Token](#))
- <FNCI_PROJECT_NAME> with the project you created earlier to capture the inventory
- <tag for your GitLab-Runner> with the tag of your GitLab runner

`%CI_PROJECT_DIR%` is the GitLab variable for the project path where the code is built. You can replace it with the path of the folder containing the binaries of your built project.

Executing the Build

The next time your build is executed, a FlexNet Code Insight agent scan is performed at the end of the build process. If you have scheduled the Code Insight scan job after a Maven build, for example, you should see something like this in your GitLab pipeline:



4

Integrating with Source Code Management

The following topics are covered in this section:

- [Why Use Source Code Management \(SCM\)?](#)
- [Configuring SCM](#)
- [SCM Command Line Client](#)
- [Git Protocol Options](#)
- [Perforce Protocol Options](#)
- [TFS Protocol and Credentials Configuration](#)

Why Use Source Code Management (SCM)?

To support deep scanning, it is necessary to bring the project codebase files to the scan server. FlexNet Code Insight provides the following ways to bring codebase files into the system:

- **Upload a codebase into FlexNet Code Insight:** Uploading a codebase is useful to analysts who typically perform ad-hoc scans on an arbitrary snapshot of code provided by the product team.
- **Use a version control SCM connector:** SCM connectors provide an automated way to fetch the code based on criteria, such as build, release, calendar, checkin, and other information. SCM connectors support various authentication mechanisms, including anonymous, username and password, and token/key/ticket on a scan server.

Configuring SCM

FlexNet Code Insight supports SCM connectors to allow remote codebases to be obtained before a scan.

Prerequisites

The following are prerequisites for using Code Insight SCM connectors:

- [SCM Command Line Client](#)
- [Recommended Clients](#)
- [Setting the Environment Variable](#)

SCM Command Line Client

Before you proceed, ensure that an SCM command-line client is installed and configured on the FlexNet Code Insight Scan Server as this is necessary for FlexNet Code Insight to be able to connect and sync to an SCM repository.

To verify that the SCM client is installed and available to FlexNet Code Insight, open a command prompt and navigate to the FlexNet Code Insight root directory. Execute a command specific to your SCM, such as:

- `git help`
- `p4 help`
- `tf help`

If the system cannot find the command specified, verify that the SCM client directory is part of the PATH variable on this server. Consult your SCM documentation for more information on how to install and configure the client.

Recommended Clients

The following is a list of clients known to work well with FlexNet Code Insight:

SCM	Client	Cost	Download Site
Git	Git	Free	http://git-scm.com/downloads
Perforce	Perforce	Paid	https://www.perforce.com/downloads
Team Foundation Server (TFS)	Team Explore Everywhere Command Line Client (TEE-CLC)	?	https://github.com/Microsoft/team-explorer-everywhere/releases



Note • Download site links are subject to change.

TEE-CLC Requirement for a TFS Connection

TEE-CLC is the TFS client required by Code Insight to connect to and synchronize with an TFS collection. Once this client is installed on the same machine where the Code Insight scanner resides, run the following command to accept the end-user license agreement:

```
tfs -eula
```

If Code Insight attempts to connect to TFS before this command is run, the connection fails.

Setting the Environment Variable

If you run the SCM command line client from a Windows machine, add your SCM client location to the PATH environment variable.



Note • Your SCM may require other environment variables to be set. Consult your SCM documentation.



Task

To set the environment variable, do the following:

1. To find your PATH environment variable settings, navigate to **Control Panel > System > Advanced System Settings**.
2. Click **Environment Variables**.
3. Look for the PATH system variable and make sure that it is set to the location of your SCM bin directory.
4. If you edit the system variable, ensure that you save your changes.

Git Protocol Options

Git repositories reside on public servers, such as GitHub and Bitbucket, or on Git servers within a corporate network. The Git URL used to clone the repository into your SCM destination folder will vary depending on your desired protocol. The following are the available protocol options.

- [Anonymous HTTP](#)
- [Authenticated HTTP](#)
- [SSH](#)
- [SSH Over HTTP](#)

Anonymous HTTP

This protocol can be used for a public repository. Public repositories can be cloned without providing an account and password.

Type	Example
GitHub Example	<code>http://github.com/myacct/Spoon-Knife.git</code>
Bitbucket Example	<code>http://bitbucket.org/myacct/myquotefork.git</code>

Authenticated HTTP

This protocol can be used for a private repository. Provide an account and password as shown in the URL format below. Use a colon between the account and password.

Type	Example
GitHub Example	<code>https://myacct:password@github.com/myacct/Hello-World.git</code>
Bitbucket Example	<code>https://myacct:password@bitbucket.org/myacct/bb101repo.git</code>

SSH

This section describes SSH authentication between a system running FlexNet Code Insight and Git servers such as GitHub and Bitbucket. The following options are possible:

- Use one SSH keypair for all Git servers.
- Use a separate keypair for each Git server.
- Use multiple keypairs for some or all Git servers.

SSH does not rely on account passwords but rather on a pair of keys, one a private key and the other a public key. Though a private key file may be protected by a password, no password should be specified for private keys used by FlexNet Code Insight.

Creating Keypairs

Use `ssh-keygen` to create a keypair for each Git server. Press **Return** twice to make the passphrase empty. For example:

```
ssh-keygen -f ~/.ssh/id_rsa_github_test1 -C "github test 1"
ssh-keygen -f ~/.ssh/id_rsa_bitbucket_test1 -C "bitbucket test 1"
```

The following files are created:

Type	Private Key	Public Key
GitHub	<code>id_rsa_github_test1</code>	<code>id_rsa_github_test1.pub</code>

Type	Private Key	Public Key
Bitbucket	id_rsa_bitbucket_test1	id_rsa_bitbucket_test1.pub

The private keys remain in the `.ssh` folder on Linux or the `<user_home>\.ssh` folder on Windows. Each public key will be stored on a Git server.

Adding to the Config File

Update `.ssh/config` (on Linux) or `<user_home>\.ssh\config` (on Windows):

Property	Github	Bitbucket
Host	github.com	bitbucket.org
User	git	git
HostName	github.com	bitbucket.org
PreferredAuthentications	publickey	publickey
IdentityFile	~/.ssh/id_rsa_github_test1	~/.ssh/id_rsa_bitbucket_test1

There is a correspondence between the name on the Host line and the name used in the URL. When there is only one keypair per host, it is convenient to specify Host as above. This means the URL for git clone is:

```
git clone git@github.com:account/repository.git
```

The following definitions allow multiple keys to be used with GitHub or Bitbucket:

Property	Github 1	Github 2
Host	mygithub_01	mygithub_02
User	git	git
HostName	github.com	github.com
PreferredAuthentications	publickey	publickey
IdentityFile	~/.ssh/id_rsa_github_test1	~/.ssh/id_rsa_github_test2

The URLs are changed to use the values of Host from the config file. The following are the appropriate git clone commands:

```
git clone git@mygithub_01:account/repository.git
git clone git@mygithub_02:account/repository.git
```

Both clone commands will connect to github.com, which is the value of **HostName**. The first command will use the private key `id_rsa_github_test1`. The second command will use the private key `id_rsa_github_test2`.

SSH Over HTTP

The standard SSH port is 22. To run SSH over port 443, perform the steps in the [SSH](#) section. The only difference is in `.ssh/config` on Linux or `<user_home>\.ssh\config` on Windows. The following are some examples:

Property	Example 1	Example 2
Host	github.com	gitssh-https
User	git	git
Port	443	443
HostName	ssh.github.com	ssh.github.com
PreferredAuthentications	publickey	publickey
IdentityFile	~/.ssh/id_rsa_github_test1	~/.ssh/id_rsa_github_test1
URL	git@github.com:account/repository.git	git@gitssh-https:account/ repository.git

Perforce Protocol Options

Perforce depots reside on an enterprise server. You have the following protocol options.

- Authenticated TCP
- Authenticated SSL

For details on how to configure the Perforce SCM instance, refer to the *FlexNet Code Insight User Guide*.

TFS Protocol and Credentials Configuration

The following describes configuration you might need for Code Insight synchronization with TFS:

- [HTTPS Protocol Support](#)
- [Special Requirement for VSTS Projects in TFS](#)

HTTPS Protocol Support

HTTPS is supported for communication between Code Insight and TFS. Perform the following steps to enable the SSL configuration for HTTPS.



Task

To enable SSL configuration:

1. Export the Secure Site SSL certificate from the browser location (shown here) for the given TFS instance:
`https://<TFS-Host>/tfs/DefaultCollection/<Project>`
2. Import the certificate in the Java (JRE) keystore, using the following command (replacing `tfs.cer` with the actual certificate file name). The certificate should be imported to the same location where the TEE-CLC and Code Insight scanner reside (see [TEE-CLC Requirement for a TFS Connection](#)).

```
keytool -import -trustcacerts -keystore cacerts -storepass changeit -noprompt -alias tfs -file  
tfs.cer
```

Special Requirement for VSTS Projects in TFS

If Code Insight is synchronizing with a VSTS (Visual Studio Team Services) project in TFS, alternate VSTS authentication credentials are required for the synchronization.



Task

To enable alternate authentication credentials needed for Code Insight synchronization with a VSTS project in TFS:

1. In Visual Studio, enable a set of alternate authentication credentials. (See the Visual Studio documentation for instructions.)
2. Specify these alternate credentials for the **Username** and **Password** in the TFS SCM instance configuration in Code Insight. See [Adding a TFS SCM Instance to the Code Insight Project](#) in the “Configuring Source Code Management” chapter in the *FlexNet Code Insight User Guide*.

Integrating with Application Lifecycle Management

This chapter covers the following topics:

- [About Integration with Application Lifecycle Management \(ALM\) Systems](#)
- [The Jira Plugin](#)

About Integration with Application Lifecycle Management (ALM) Systems

FlexNet Code Insight support application lifecycle management (ALM) system plugins to manage external work items. These plugins allow Code Insight users to create and manage external work items associated with inventory in the ALM system directly from Code Insight so that inventory requiring further review and remediation can be tracked externally as part of the user's existing issue tracking system.

For example, a Code Insight scan might uncover security vulnerabilities or copyleft licenses requiring further review by the Security and Legal teams. With an ALM integration, these issues can be quickly converted into work items that point to corresponding issues in the ALM instance. The plugins support pre-populated data and a synchronization of data between Code Insight and the server based on a configured synchronization frequency.

Currently, Code Insight offers a Jira plugin (see the next section, [The Jira Plugin](#)). Future releases will provide additional integrations with other ALM systems.

The Jira Plugin

The Jira plugin provided by Code Insight can be used to create new Jira work items directly from Code Insight. These work items allow management of external remediation work associated with inventory items in Code Insight.

The following sections describe how to configure the Jira plugin for Code Insight integration with your Jira instances:

- [Prerequisites for the Jira Plugin](#)
- [Configuring the Jira Plugin](#)

Prerequisites for the Jira Plugin

The Jira plugin is included with Code Insight, is located on the core server in the `config/core/plugins` directory. Ensure that this directory contains the latest Jira plugin, particularly after migrating to the latest Code Insight version.

Additional requirements include the following:

- The Jira plugin requires access to a Jira server with credentials for a valid user on this server. The designated user will be used to authenticate Code Insight on the Jira server and will also be listed as the reporter on the issue created from Code Insight.
- The specified user must have full access to the Jira instance, particularly if Captcha or Single Sign-On are enabled on the Jira server.

You can use the **Test Connection** button on the ALM configuration page for the Jira instance to validate a successful connection to the Jira server. (See [Adding a Jira Instance](#) in the next section, *Configuring the Jira Plugin*.)

Configuring the Jira Plugin

The Jira plugin can be configured to connect to multiple Jira instances and to display default values for each field in the configured instance. Projects can then be individually assigned to connect to and synchronize to one the configured instances.

The following topics describe how to configure and maintain a Jira instance:

- [Adding a Jira Instance](#)
- [Using Code Insight Variables](#)
- [Synchronizing Work Items](#)
- [Deleting an ALM Instance](#)

Adding a Jira Instance

The system Administrator can configure one or more Jira instances and their default field values globally at the application level using the **Administration** menu. Once configured, the Jira instances are available in the **Edit Project** section so that they can be associated to a specific project.



Task

To add a Jira instance:

1. As system Administrator, select **Administration** from the main menu.
2. Select the **ALM** tile on the left.
3. Select **Jira** from the **Application** dropdown list.
4. Click **Add Instance**. The **Instance** configuration tab is displayed.
5. Enter values for the required fields based on your Jira server information. The following fields are required. (See the inline help for explanations of the fields.)
 - **ALM Instance Name**

- **JIRA Server URL**
 - **JIRA Username**
 - **JIRA Password**
6. Once you have completed the required fields, click the **Test Connection** button on the right to validate that Code Insight can connect to the specified Jira server.

If the connection is successful, a “connection successful” message is displayed. Otherwise, reenter the credentials and try again. Ensure that the specified user has full access to the Jira instance, particularly if Captcha or Single Sign-On are enabled on this Jira server.
 7. Complete the remaining fields. See the inline help for explanations of the fields.

You can include inventory variables in the **Default Summary** and **Default Description** fields that will be replaced by actual values in the newly created Jira issue and work item. For a list of supported variables, see the next section, [Using Code Insight Variables](#).
 8. Click **Save** to save the Jira instance. The Jira sever settings and mandatory values are validated.

Using Code Insight Variables

The **Default Summary Text** and **Default Description Text** fields support Code Insight variables that can communicate details about the Code Insight project, inventory item, and other relevant information in the work item and associated Jira issue.

Supported Variables

The following table lists the available variables for use in the text entered in the **Default Summary Text** and **Default Description Text** fields:

Table 5-1 • Supported Code Insight Variables For Use in Work Item Summary and Description Text

\$PROJECT_NAME	Name of the Code Insight project containing the issue
\$INVENTORY_ITEM_NAME	Name of the inventory item containing the issue
\$COMPONENT_NAME	Name of the component associated with the inventory item
\$VERSION_NAME	Version of the component associated with the inventory item
\$LICENSE_NAME	Name of the selected license for the inventory item
\$NUMBER_VULNERABILITIES	Total number of security vulnerabilities associated with the inventory item
\$NUMBER_FILES	Total number of files associated with the inventory item
\$INVENTORY_URL	Link to the inventory item

When the work item is created, the included variables are replaced by their respective values.

Example Use of Variables

The following is example text you might enter in the **Default Summary Text** field. The text includes some of the available variables:

The `$INVENTORY_ITEM_NAME` inventory item in the project `$PROJECT_NAME` contains `$NUMBER_VULNERABILITIES` vulnerabilities that require review. Go to `$INVENTORY_URL` to see the vulnerable inventory item.

If your Code Insight project name is *MySampleProject* and the name of the inventory item name for which you create a work item is *Apache Commons BeanUtils*, the work item and Jira issue will display the following summary:

The Apache Commons BeanUtils 1.7.0 (Apache 2.0) inventory item in the project MySampleProject contains 18 vulnerabilities that require review. Go to <https://my.sample.server:8888/codeinsight> to see the vulnerable inventory item

Synchronizing Work Items

FlexNet Code Insight provides the ability to synchronize work items between Code Insight and the ALM system so that Code Insight always reflects the most current state of each work item. The one-way synchronization updates the following fields for the work item in Code Insight: **Status, Type, Priority, Assignee, Summary**.

The following procedure describes how to set the frequency of this synchronization process (labeled **Existing Issues Sync Frequency** on the ALM tab).



Note • The Sync Frequency configuration applies to all the ALM instances. If not explicitly set, the sync frequency defaults to Daily.



Task

To configure the issue sync frequency:

1. As system Administrator, select **Administration** from the main menu.
2. Click the **ALM** tab.
3. Click the **Edit Sync Frequency** icon on the right (to the right of the **Existing Issues Sync Frequency** value).
4. Select one of the frequency options—**Never**, **Hourly**, **Daily**, or **Weekly**—and complete their respective sub-options.
5. Click the **Save Changes** icon to save or **Cancel** to discard the setting.

Work Item Status Updates

If the status of the work item in the ALM system changes, the status of the work item in Code Insight will reflect the change after the synchronization completes. This can result in a change to the # **Open Work Items** and # **Closed Work Items** for each inventory item. These links and the **Open Work Items** information alert link will be updated to reflect the change. Additionally, the **Inventory with Open Work Items** selection in **Advanced Search** may return a different number of results.

The following lists the default status values:

- The default Open status values for Jira include **Open**, **Reopen**, **New**, **To Do**, **In Progress**, and **Backlog**.
- The default Closed status values for Jira include **Done**, **Resolved**, **Verified**, and **Closed**.

Custom statuses are not currently supported.

Deleting an ALM Instance

The application Administrator can delete an ALM instance as long as no projects currently reference the instance.

If the instance that you want to delete is referenced by a project, it cannot be deleted until the instance is unassociated from the project. See the *FlexNet Code Insight User Guide* for instructions on how unassociate an instance from a project.



Task

To delete an ALM instance:

1. As the system Administrator, select **Administration** from the main menu.
2. Select the **ALM** tab.
3. Select the **Instance** tab for the instance you want to delete.
4. Click the **Delete Instance** button.

